VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Martin Elis

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

DATOVĚ ÚSPORNÉ ZABEZPEČENÍ CLOUDOVÝCH ÚLOŽIŠŤ
DATA-EFFICIENT SECURITY OF CLOUD STORAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Martin Elis**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Vlastimil Člupek**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Martin Elis ***ID:*** 110893
Ročník: 2 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Datově úsporné zabezpečení cloudových úložišť

POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce proveďte hloubkovou analýzu současných možností zabezpečení cloudových úložišť a popište možné útoky na ně. Na základě provedené analýzy navrhněte zabezpečení mezi uživatelem a cloudem s využitím dostupných kryptografických primitiv. Funkčnost navrženého schématu ověřte praktickou realizací ve vhodném prostředí. Dále proveďte srovnání výpočetní a datové náročnosti navrženého schématu v závislosti na použití různých délek kryptografických klíčů a různých algoritmů ze symetrické a asymetrické kryptografie a dále u každé varianty proveďte ohodnocení síly kryptografického zabezpečení a uveďte příklad možného nasazení v praxi.

DOPORUČENÁ LITERATURA:

[1] MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. Handbook of applied cryptography. CRC press, 1996.

[2] VELTE, Anthony T, Toby J VELTE a Robert C ELSENPETER. Cloud Computing: praktický průvodce. Vyd. 1. Brno: Computer Press, 2011, 344 s. ISBN 978-80-251-3333-0.

[3] MOHAMMAD, Jasim, et al. Securing Cloud Computing Environment using a New Trend of Cryptography. In: Cloud Computing (ICCC), 2015 International Conference on. IEEE, 2015. p. 1-8.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 25.5.2016

Vedoucí práce: Ing. Vlastimil Člupek
Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Tato práce je zaměřena na cloudovou problematiku, zejména pak na její bezpečnostní stránku. Popisuje současné bezpečnostní trendy a přístupy používané bezpečnostními techniky při tvorbě sofistikovaných návrhů zabezpečených cloudových systémů. Součástí je analýza rizik a přehled nejčastějších typů útoků vedených vůči cloudovým řešením. Rovněž se tento dokument zabývá možnostmi, principy, výhodami a negativy jednotlivých cloudových distribucí. Další text se zabývá obvyklými užívanými metodami přístupu ke cloudu. Práce obsahuje autorův vlastní návrh možné realizace. V další části práce je pak zdokumentován samotný postup výstavby cloudového úložiště a principy jeho zabezpečení. V závěru práce se autor věnuje srovnání šifrovacích algoritmů a jejich chování v závislosti na použitých délkách klíčů.

KLÍČOVÁ SLOVA

Algoritmus, autentizace, autorizace, bezpečnost, cloudový výpočet, integrita, důvěrnost, šifrování ,virtualizace, virtuální privátní síť.

ABSTRACT

This work is focused on problematics of a cloud solution, especially on its security side. It describes the current security trends and approaches used by security engineers when creating sophisticated designs of secure cloud systems. As part of it there is a risk analysis and an overview of the most common types of attacks led against the cloud solutions. Also, this document deals with the possibilities, principles, advantages and negatives of different types of cloud distributions. Another text deals with the usual methods used for accessing the cloud. This thesis contains author's own design of possible realization. In the next part of the document, process of building a safe cloud data storage is described together with principles of ensuring its security. In the conclusion, the author focuses on comparison of cryptographic algorithms and their behavior depending on the length of a used keys.

KEYWORDS

Algorithm, authentication, authorization, security, cloud computation, integrity, confidentiality, encryption, virtualization, virtual private network.

ELIS, M. *Datově úsporné zabezpečení cloudových úložišť*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 69 s., 2 s. příloh. Diplomová práce. Vedoucí práce: Ing. Vlastimil Člupek

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma *Datově úsporné zabezpečení cloudových úložišť* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Vlastimilu Člupkovi za odborné vedení, konzultace a podnětné návrhy k práci. Rovněž bych chtěl poděkovat zaměstnavateli, společnosti Xura Czech Republic s.r.o. za umožnění realizace praktické části práce na jejich systémech. V poslední řadě bych chtěl poděkovat rodině a přítelkyni, jež mi byli vždy oporou a měli se mnou trpělivost.

V Brně dne ....

(podpis autora)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Purkynova 118, CZ-61200 Brno, Czechia

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

V Brně dne ....

(podpis autora)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tato diplomová práce se věnuje konkrétnímu datovému modelu, takzvanému cloudu, jeho zabezpečením před běžnými útoky a metodami přístupu k tomuto typu datových úložišť. Můj zájem o toto téma vychází z faktu, že se v daném oboru pohybuji již čtvrtým rokem a s cloudovými technologiemi přicházím do styku denně. Tato zkušenost mi, jak doufám, umožní v této práci uplatnit nabyté poznatky a zprostředkuje tak čtenáři ucelený pohled na problematiku cloudových úložišť. Z praxe telekomunikačního technika tak vím, že zajištění bezpečnosti cloudového úložiště, zvláště pak úložiště využívaného pro telekomunikační účely je pro poskytovatele cloudových řešení zcela zásadní. Toto zabezpečení bývá rovněž vyžadováno a ovlivňováno nejen samotnými provozovateli cloudu, ale i jejich zákazníky, potažmo jejich uživateli. Díky tomuto se pak dá cloudové zabezpečení chápat jako komplexní soubor bezpečnostních politik, použitých technologických zařízení, metod přístupu, procesů a technik zaručujících danou úroveň zabezpečení.

Z těchto důvodů, tak toto téma zahrnuje různé oblasti počítačového zabezpečení, od technologických, přes organizační a administrativní po procesní. V praxi jsou pak tyto soubory pravidel a technik kontrolovány a ověřovány bezpečnostními firmami, které v případě splnění daných bezpečnostních standardů, udělují provozovateli cloudu adekvátní certifikáty. Tyto certifikáty garantují uživatelům daného cloudu konkrétní míru zabezpečení.

Jak již bylo výše zmíněno, součástí této práce je i analýza bezpečnostních rizik a přehled běžných útoků, jimž jsou cloudová úložiště vystavena. Ve čtvrté kapitole jsem se zabýval metodami přenosu dat mezi uživatelem a cloudovým úložištěm a jejich zabezpečením. Na základě těchto znalostí jsem vytvořil návrh cloudového přenosu, který byl v rámci diplomové práce realizován.

Součástí práce tak je kapitola šestá, věnující se popisu použité cloudové platformy, důvodům jejího výběru a výhodám daných principů a přístupů použitých v samotné realizaci. V této kapitole je rovněž popsán použitý operační systém a jeho výhody ve srovnání s jinými dostupnými variantami.

Kapitola sedmá je věnována zabezpečení serverů cloudového úložiště a konfiguraci síťových prvků. Jsou zde vysvětleny principy užívané v praxi a ukázány příklady síťového nastavení, jež bylo použito v této práci. Rovněž je zde vysvětlen způsob, jakým je umožněn přístup uživatelů k serverům v cloudu.

Samotná realizace přístupu uživatele k serverům pomocí VPN klienta je pak podrobně popsán v kapitole osm. Rozveden je pak použitý software a princip připojení a autentizace uživatele vůči autentizačnímu serveru. Součástí této sekce je i podkapitola, zabývající se bezpečným uchováváním přihlašovacích údajů.

V deváté kapitole se pak zabývám použitými metodami zajištění efektivnosti ukládání dat v cloudovém prostředí. Vysvětlen je zde princip takzvaného „tlustého“ provisioningu a popsány principy souboru metod OpenDedup.

V závěru se práce zaobírá samotným přenosem souborů mezi cloudovými servery a závislostí na použitém šifrovacím algoritmu a délce šifrovacího klíče. V této sekci tak

popisuji způsob testování efektivnosti šifrovacích algoritmů, podobu testovacích dat a použité prostředky realizující samotný šifrovaný přenos. Součástí této kapitoly tak je přehledná tabulka dosažených výsledků a vyhodnocení jednotlivých variant.

1 CLOUD COMPUTING

Abychom mohli dále pokračovat a zabývat se hlouběji problematikou cloudu, respektive jeho zabezpečením, musíme nejprve pojem cloud vysvětlit. Cloud, potažmo cloudový výpočet je v současné době chápán jako model vývoje a používání výpočetních technologií uložených na internetu. Rovněž lze říci, že se jedná o typ poskytování služeb a programů uložených na webu [1]. Obvykle pak uživatelé přistupují ke svým datům, službám či programům prostřednictvím speciálního klienta nebo pomocí grafického rozhraní dostupného přes webový prohlížeč. Tohoto datového modelu dnes využívají firmy pro provoz svých aplikací, které jsou tak dostupné uživateli odkudkoli. Tento způsob poskytování služeb je tak jakýmsi kompromisem, mezi webovým a desktopovým řešením. Uvažujeme-li současné možnosti a rychlosti internetového připojení, tak cloudové řešení nabízí srovnatelnou portabilitu a interaktivitu, jaké je možné dosáhnout u webové či desktopové aplikace. Oproti desktopovým řešením pak nedochází ke spotřebě výpočetních zdrojů na straně uživatele (CPU, paměť), protože výpočet probíhá na straně cloudu. Z toho vyplývá patrná výhoda cloudového řešení, tedy nižší nároky na výpočetní zdroje uživatele, tudíž větší dostupnost aplikace pro širokou veřejnost. Stejný trend lze pozorovat i v přemýšlení velkých telekomunikačních a softwarových firem. Využití cloudového řešení tak umožňuje firmám snížení nákladů o částku, kterou by jinak musela investovat do hardwarového vybavení (CAPEX – takzvané kapitálové výdaje), jeho údržby a školeného technického personálu (OPEX – takzvané operační výdaje). Rovněž škálovatelnost a rozšiřitelnost vyplývající z možnosti rychle navýšit množství výpočetní kapacity (zdrojů) překonává výše zmíněné modely v mnoha směrech [2]. Mezi další nesporné výhody cloudu patří možnost aktualizace nabízených služeb. V tomto případě se o aktualizaci samotnou nestará koncový uživatel, ale poskytovatel služby. Nabídka aplikací v cloudovém prostředí je velmi pestrá, od kancelářských balíků, softwarová a vývojová prostředí, přes systémy pro distribuovaný výpočet, až po hostování virtuálních strojů, nesoucí operační systémy. Komunikace s takovýmito aplikacemi, či stroji, je opět možná pomocí webových grafických rozhraní, či konzolí. Aplikaci v cloudu jako takovou si lze například představit jako server, anebo cluster serverů, nakonfigurovaných za účelem provádění nějaké akce či poskytování konkrétních funkcí uživateli. Základním principem cloudu je pak virtualizace. Pojmem virtualizace rozumíme využití a přidělení výpočetních zdrojů nezávisle na jejich fyzické reprezentaci a hardwaru. Konkrétní rozložení zdrojů zůstává abstraktní koncovému uživateli. Virtualizace nám tak umožňuje snadněji disponovat dostupnými zdroji a vytvářet tak specializované virtuální stroje schopné realizovat danou funkci, bez nutnosti využití specifického hardwaru.

1.1 Typy cloudu podle nasazení

Existují různé pohledy na cloud a různá členění. Z pohledu infrastruktury tak rozlišujeme tři hlavní typy cloudů, veřejný, privátní a hybridní.

1.1.1 Privátní cloud

Jak už samotný název napovídá, jedná se o privátní cloudové řešení poskytované konkrétnímu uživateli (organizaci). Za správu takového cloudu zodpovídá interně uživatel, anebo poskytovatel cloudového řešení. Samotné datacentrum pak může být opět hostováno interně nebo poskytovatelem. Tento typ cloudového řešení může vést k vysoké míře zefektivnění, díky sdílení výpočetních zdrojů, avšak značnou nevýhodou je vysoká náročnost na virtualizaci, její plánování a související možnost pro vznik bezpečnostních rizik. V neprospěch rovněž hovoří vysoká cenová náročnost vyplývající z pořizovací ceny hardwaru, nároky na prostor a náklady na údržbu [3].

1.1.2 Veřejný cloud

Oproti privátnímu cloudu, cloud veřejný zprostředkovává své služby široké veřejnosti pomoci veřejně přístupné sítě. Z pohledu použitých technologií a architektury se tak zmíněné cloudy příliš neliší, avšak potencionální bezpečnostní rizika spojené s přenosem dat a informací pomocí nedůvěryhodného spojení jsou diametrálně odlišná. Případným útočníkům se zde nabízí obrovský prostor pro útoky. Tyto mohou být zacíleny na cloudové aplikace, servery, či infrastrukturu samotného cloudu. Kromě již zmíněného přístupu z veřejného internetu mnozí cloudoví poskytovatelé využívají dedikovaných privátních spojení, která umožňují zákazníkovi připojení ke konkrétnímu přístupovému bodu v cloudu. Příkladem těchto spojení mohou být například Cisco AnyConnect, Direct Link firmy IBM, Azure ExpressRoute, či Amazon Web Service Direct Connect [4].

1.1.3 Hybridní cloud

Spojením alespoň jednoho privátního a jednoho veřejného cloudu dostáváme cloud hybridní. Tento typ cloudu disponuje vlastnostmi obou předešlých typů cloudů, přičemž zpřístupňuje uživateli rozšiřující možnosti navýšení výpočetní kapacity, provázaní se systémy třetích stran, agregaci, integraci a „kustomizaci“ jednotlivých cloudů. Tento model také nabízí největší míru škálovatelnosti a flexibility. Rovněž umožňuje zavést politiky založené na principu oddělování citlivých dat od ostatních. V takovém případě pak udržujeme citlivá data v privátním cloudu, zatímco méně zranitelná data či aplikace jsou dostupná v cloudu veřejném a odtud jsou dále poskytovány koncovým uživatelům.

1.1.4 Další typy cloudů

V praxi můžeme nalézt ještě další typy cloudových řešení, avšak lze všeobecně říci, že se jedná o variace již zmíněných typů cloudů, tudíž se jimi nebudu v této práci zabývat [5].

1.2 Typy cloudu podle distribučního modelu

Jedním z nejrozšířenějšího členění cloudových služeb je pak rozdělení odvozené dle způsobu nabízené služby - distribučního modelu. Distribučním modelem v tomto případě chápeme způsob, jakým nabízejí „cloudoví provideři“ svou službu. Obvykle tedy infrastrukturu, software nebo platformu.

1.2.1 Infrastruktura jako služba

IaaS (Infrastructure as a Service) – infrastruktura jako služba. Jedná se o nejjednodušší model cloudové služby, kdy se cloud provider zavazuje zákazníku poskytovat servery, fyzické HW servery, či stále obvyklejší virtualizované stroje a další potřebné výpočetní zdroje. Tento model tak představuje situaci, kdy uživatel má k dispozici webově dostupnou službu, jejíž skutečná fyzická realizace mu je zcela abstrahována. Uživatel tak nemá informace o používaných výpočetních zdrojích, zabezpečení dat, lokací, či pohybem, rozložením a zálohou dat. Daná webově dostupná služba je pak vykonávána specializovanými servery, virtuálními stroji, které běží na standartních hardwarových strojích jako „hosti“. Tito hosté jsou kontrolováni takzvaným „hypervisorem“. Nejznámějšími a komerčně nejúspěšnějšími hypervisory jsou například VirtualBox od společnosti Oracle, KVM, či Vmware ESX/ESXi [6]. Součástí nabízené infrastruktury tak většinou jsou nejrůznější firewally, směrovače, přepínače, datová úložiště, load-balancery a další síťová zařízení (VLAN, SAN). Tato zařízení se pak nacházejí v datových centrech. Tato datová centra musejí splňovat nejrůznější bezpečnostní požadavky zajišťující ochranu dat a fyzických strojů před neoprávněným přístupem. Tato bezpečnostní opatření jsou zajištěna nejrůznějšími fyzickými zařízeními (například bezpečnostní dveře, snímače otisků prstů, přístupové karty, zámky a přístupové kombinace, ostraha, kamerové systémy atd.). Instalace cloudové aplikace v takovémto cloudu pak obvykle obnáší instalaci operačního systému pomocí webového rozhraní a „image“ instalované aplikace. Poplatky za takovéto hostování aplikace jsou obvykle počítány na základě vyhrazených a spotřebovaných výpočetních zdrojů (paměť, šířka pásma, velikost disku, SW licence) [7].

1.2.2 Software jako služba

SaaS (Software as a Service) - software jako služba. Tento typ cloudové distribuce poskytuje uživateli přístup k softwaru aplikace a potřebným databázím. Uživatel si tak pronajímá přístup k softwaru uloženém v cloudovém úložišti. Obvykle je tento poplatek realizován jako jednorázový poplatek při použití aplikace, či jako měsíční, či roční předplatné (v případě operátorů například součást tarifu). Příkladem takovéto služby může být Google App.

V případě SaaS poskytovatel cloudu ručí za správu infrastruktury a platforem, na kterých je aplikace provozována. Oproti předchozímu modelu, tak aplikaci do cloudu nasazuje a posléze provozuje poskytovatel cloudu, přičemž uživatelé přistupují k aplikaci pomocí cloudových klientů. Díky tomuto je eliminována nutnost instalovat a spouštět aplikace na jednotlivých zařízeních uživatelů, což zjednodušuje údržbu a snižuje vytížení techniků podpory [8].

Velkou výhodou tohoto typu cloudu je pak škálovatelnost. Jak už bylo výše zmíněno, cloudová realizace umožňuje dynamickou změnu alokovaných výpočetních zdrojů v reakci na aktuální stav spotřeby. Této dynamičnosti se obvykle dosahuje pomocí

klonování procesů a jejich redistribucí mezi množství aplikačních serverů. K rovnoměrnému rozložení zátěže na virtuální strojích se využívá takzvaných load-balancerů. Toto přerozdělování zdrojů, zátěže a procesů je uživateli transparentní. Vzhledem k množství uživatelů takovéhoto cloudu, je třeba zajistit takzvanou multitenantnost, schopnost systému obsluhovat současně velké množství uživatelů.

Nezanedbatelnou výhodou tohoto typu cloudu je pak úspora za správu cloudové infrastruktury, která tak umožňuje firmám investovat finanční prostředky jinam. Nevýhodou pak může být možnost neautorizovaného přístupu k datům zákazníka, protože se data nacházejí na serveru na straně poskytovatele. Z tohoto důvodu se bezpečnost přístupu stále častěji zajišťuje prostřednictvím ověřování klíčů třetí nezávislou stranou.

1.2.3 Platforma jako služba

PaaS (Platform as a Service) – platforma jako služba. Tento model cloudového řešení je založen na principu poskytnutí vývojového prostředí uživateli. Uživateli je tak poskytována výpočetní platforma – server, jež zahrnuje operační systém, vývojové prostředí daného programovacího jazyka, databáze a webový server. Vývojář aplikace používající takovéto prostředí může vyvíjet a provozovat svou aplikaci bez potřeby porozumění hardwarovým požadavkům [9]. Toto odstínění od architektury tak zásadně ovlivňuje a zjednodušuje vývoj jinak velmi složitých aplikací. Rovněž mu tento model umožňuje vynaložit finančních prostředky za hardwarové a softwarové vybavení jiným způsobem. V současné době cloudové modely PaaS dokáží automaticky regulovat množství přidělovaných výpočetních prostředků v závislosti na potřebách dané aplikace. Uživatel je tak ušetřen nutnosti stanovit potřebné výpočetní zdroje a současně nemusí manuálně navyšovat výpočetní kapacitu v případě změn vytíženosti provozované aplikace. Jelikož tento model poskytuje uživatelům zabudovanou infrastrukturu, jednoduchou správu a bohaté možnosti rozšíření, lze v zásadě tvrdit, že tento model celkově zefektivňuje cyklus vývoje softwaru. Toto cloudové řešení se rovněž osvědčilo v případech, kdy na aplikaci spolupracuje větší množství vývojářů přistupujícím k vývojovému nelokálně ze vzdálených lokací. Příkladem PaaS modelů mohou být služby společnosti Microsoft Azure, či Google App Engine [7].

2 ZABEZPEČENÍ CLOUDU

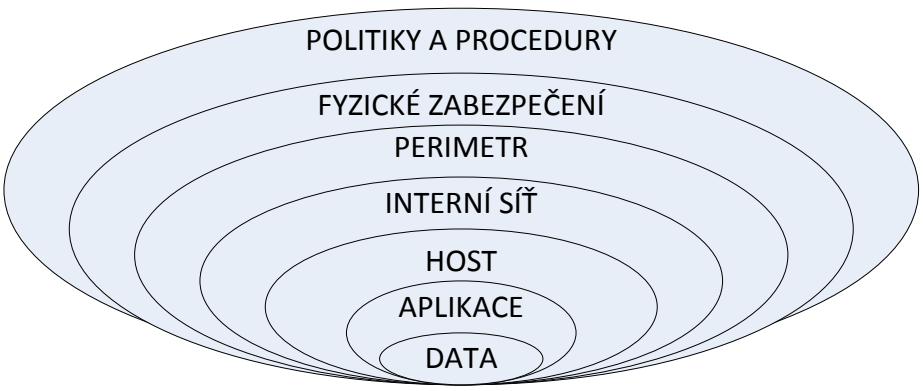
Co znamená bezpečnost? Bezpečnost je definována jako kvalita či stav bezpečí, to znamená situaci, kdy jsme chráněni před hrozbou. Bezpečnost je tedy stupeň odolnosti, či ochrana proti újmě. Aplikuje se na veškerá zranitelná aktiva, jakožto osoby, nemovitosti, společnost, národ či organizaci. V oblasti počítačových technologií to pak znamená ochranu informací ve formě dat vůči útokům vedeným na jejich klíčové elementy (důvěrnost, integritu, dostupnost, autentičnost), včetně systémů a hardwarového vybavení, které tyto data používají, uchovávají či přenáší. Ochrana informací se pak zajišťuje prostřednictvím bezpečnostních strategií jejich vhodnou kombinací. Tyto bezpečnostní strategie zahrnují bezpečnostní politiky, použité technologie, bezpečnostní připravenost personálu a programy zvyšující povědomí uživatelů o aktuálních rizicích [10].

Existuje velké množství literatury zabývající se analýzou a kategorizací bezpečnosti. Částečnou příčinou může být fakt, že ve většině bezpečnostních systémů lze identifikovat takzvaný nejslabší článek. Současně tento nejslabší prvek systému je z hlediska bezpečnosti nejdůležitější, protože představuje největší potencionální riziko narušení bezpečnosti. Záleží však, z jaké perspektivy na systém pohlížíme, a tak tímto článkem mohou být různé prvky systému.

Z pohledu bezpečnostního inženýra je navíc situace ještě více komplikovaná. Je to způsobeno tím, že skutečný stav je velmi asymetrický. Zatímco bezpečnostní inženýr musí vzít v úvahu a ošetřit veškeré napadnutelné součásti systému, útočníkovi stačí pouze identifikovat potencionální slabé místo a na to svůj útok koncentrovat.

2.1 Cibulový model bezpečnosti

V praxi se bezpečnost cloudu tedy řeší na několika úrovních. Každá z těchto úrovní přispívá celkovému zabezpečení systému technologiemi dané vrstvy, přičemž adresuje bezpečnostní rizika této vrstvy. Pro lepší představu a zjednodušení jsem využil takzvaného cibulového modelu bezpečnosti [11] na obrázku 1, který vzápětí podrobněji popíšu.



Obr. 1: Cibulový model bezpečnosti.

2.2 Přehled bezpečnostních úrovní

2.2.1 Politiky a procedury

Politiky jsou sady dokumentů, jež obsahují pravidla, nařízení a postupy používané pro přístup k datům uloženým v cloudovém úložišti. Tato dokumentace reguluje, jakým způsobem organizace spravuje, chrání a přenáší citlivé informace (vlastní i klientská data) a tvoří základ pro počítačově orientovanou síťovou bezpečnost organizace. Existuje široké spektrum možností, jak se dají definovat firemní pravidla. V praxi se užívá několik typů dokumentů a postupů definujících politiku firmy.

Politika je všeobecné, na technologii nezávislé nařízení, přesně definující oblast působnosti a požadovanou formu zabezpečení. Politiky v informačních technologiích jsou iniciovány výkonnými vedeními firem v případech, kdy situace vyžaduje zavedení politiky v reakci na aktuální dění (nové bezpečnostní hrozby, technologie).

Standard obvykle odkazuje na povinné postupy, úkony, pravidla, či regulace, jež musí být splněny při výkonu dané aktivity, či zajištění daného typu zabezpečení. Podobně jako politiky, měly by být nezávislé na použité technologii. Standardy bývají vytvářeny širokými skupinami odborníků napříč firmami. Příkladem může být standard ISO-IEC 27001 [12] popisující správu zabezpečení informací.

Průvodci obsahují doporučené postupy a rady provozovatelům, IT personálu a ostatním uživatelům v případech, kdy je konkrétní standard neaplikovatelný. Průvodci jsou vytvářeny odborníky na danou problematiku napříč firmami. Oproti předchozím dokumentům, mohou být vztahovány vůči konkrétnímu typu a verzi použité technologie. Jako příklad bych uvedl operační manuál pro konkrétní software, popisující standardní konfiguraci a operace.

Procedury jsou detailní návody a postupy, které je třeba vykonat za účelem splnění konkrétního úkolu (opatření). Velmi často jsou závislé na použité technologii a její implementaci. Procedury jsou na nejnižší úrovni v hierarchii bezpečnostní politiky. Procedury jsou obvykle vytvářeny konkrétními odděleními firem a jsou sdíleny v případě potřeby. Příkladem může být procedura pro připojení k virtuálnímu stroji v cloudu

2.2.2 Fyzická bezpečnost

Fyzické zabezpečení popisuje bezpečnostní opatření, jež jsou navržena pro zamezení neoprávněného přístupu k zařízením, vybavením a zdrojům a k ochraně zaměstnanců a majetku před poškozením, či zneužitím (špionáž, krádež, fyzické poškození) [13]. Cílem této vrstvy je rovněž rozlišit mezi oprávněními osobami a narušiteli. Současně je úkolem odstrašit potencionálního narušitele (například varování). Dalším úkolem je detekovat a monitorovat případné narušitele, včetně adekvátní odezvy (bezpečnostní služba, policie).

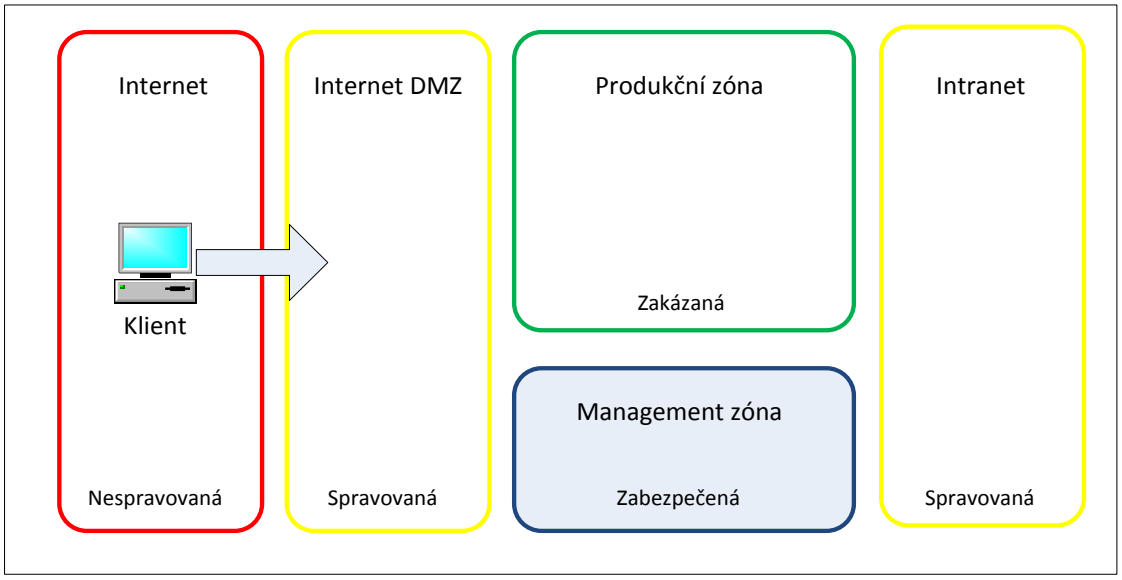
Návrh této úrovně zabezpečení je pak záležitostí bezpečnostních architektů a analytiků, kdy musejí uvážit prvky zabezpečení vzhledem k možným rizikům, přičemž musejí kalkulovat s náklady na specifikaci, vývoj, testování, implementaci, používání, správu, monitorování a údržbu bezpečnostních prvků. Rovněž je třeba v plánování fyzického zabezpečení zahrnout faktory, jako jsou lidská práva, estetika, zdraví a bezpečí, sociální normy a konvence. V případě cloudového řešení zodpovídá za zabezpečení této vrstvy výhradně poskytovatel.

Mezi obvyklé metody této vrstvy patří přístupové karty, kamerové systémy, bezpečnostní služby, bezpečnostní osvětlení, zámky, přístupové protokoly, zdi, ploty, zábrany a další.

2.2.3 Bezpečnost perimetru

Perimetr je definován jako oblast či zóna, kde se nachází výpočetní technika. V minulosti nebylo nutné tuto vrstvu oddělovat od vrstvy fyzické avšak v současnosti, kdy soudobé technologie umožňují a široce podporují metody vzdáleného přístupu, je třeba považovat tyto vzdálené místa jako součást zóny. V současné době je tedy možné považovat celou korporátní síť jako perimetr [14]. Jelikož však současné sítě bývají velmi dynamické, je nutné využívat technologií, které dokáží adekvátně reagovat na změny topologie. Proto je nutné zajistit kontinuální sledování sítě, jež by mělo zaručit odhalení nepatřičného používání, či zneužívání sítě a jejich výpočetních zdrojů.

V praxi se tak síť skládá z několika odlišných částí, na něž je třeba aplikovat zásady a bezpečnostní prvky odpovídající jejich potencionální zranitelnosti. Příkladem tohoto rozdělení na zóny budiž následující ilustrativní obrázek.



Obr. 2: Příklad síťových zón.

Mezi metody zajišťující tuto vrstvu zabezpečení patří technologie VPN [15] pro externí přístup, přičemž musí být zajištěna schopnost identifikovat, kdo se snaží získat přístup (bezpečnostní tokeny, dedikovaní VPN klienti). Využívá se zde generátorů klíčů, které umožňují vytvářet unikátní hesla pro jednotlivá VPN spojení. V souladu s předchozí vrstvou se pak uplatňují metody, jakými jsou správa přístupu k serverům, zákazy vnášení elektronických zařízení do prostor serverů (USB disky, MP3 „playery“, mobilní telefony). Dále se využívá „autentizačních hubů“ (ověřují totožnost uživatele) a systémů oprávnění uživatelů k provádění operací (autorizace). S autentizací a autorizací úzce souvisí definování rolí, které udávají k jakým operacím je uživatel oprávněn. Jelikož však korporátní sítě mohou nabývat obrovských rozsahů, obvykle se přistupuje k rozdělení na menší úseky, přičemž jednotlivé části je pak jednodušší spravovat.

2.2.4 Bezpečnost sítě

Síťovou bezpečností se označuje ochrana většího množství počítačů a zařízení, které jsou navzájem propojeny sítí a současně tento pojem zastřešuje politiky a procedury implementované síťovým administrátorem za účelem zamezení a vyšetření neautorizovaného přístupu, zneužití, modifikace, či odepření přístupu k síti a síťovým zdrojům. V praxi to znamená, že dobře implementovaná síťová bezpečnost chrání síť a její součásti před útoky virů, malwaru, napadeními hackery a jinými hrozbami. Ochrana tak musí zahrnovat jak odolnost vůči neoprávněnému přístupu k chráněným prvkům, tak opatření vůči jejich modifikaci.

Cíly této vrstvy jsou tedy ochrana a zaručení důvěrnosti, integrity, autentičnosti a dostupnosti síťových zdrojů. Důvěrnost zajišťuje, že je informace dostupná pouze oprávněnému uživateli. Požadavek na integritu spočívá v nutnosti zajistit úplnost dat. Autentičnost pak zastupuje původnost a věrohodnost dat. Dostupnost pak představuje vlastnost dat býti uživateli dostupné v případě jejich vyžádání [16].

Autentizace je proces, kdy se ověřuje identita osoby či zařízení. Pomocí elektronických podpisů se pak zaručuje nepopiratelnost autorství. Díky této nepopiratelnosti jsme jednoznačně schopni identifikovat autora dat.

Tyto bezpečnostní požadavky se v informačních technologiích nejčastěji implementují pomocí firewallů. Tyto firewally slouží k pokročilému filtrováním síťového provozu, přičemž základním pravidlem je princip povolení pouze nutného síťového provozu. Ostatní provoz je implicitně zakázán a zahozen. Tím se rovněž dosáhne účinného blokování SPAMu.

Dalším hojně používaným bezpečnostním prvkem je využití systémů a technologií pro kontrolu přístupu. Standardem cloudové komunikace tak jsou VPN tunely. Rovněž se využívá takzvaných access listů a autentizačních a autorizačních serverů.

Nutností je rovněž monitorování a inspekce provozu na síťových prvcích. K tomuto se v praxi používá protokol NetFlow v případě zařízení společnosti Cisco, či protokol sFlow firmy HP [17]. Podobně jako v případě zabezpečení perimetru i na této vrstvě se využívá rozdělení na zóny a podzóny. Platí však pravidlo znemožnění přímého přístupu mezi jednotlivými zónami. Mezi technologie zabezpečení na této vrstvě rovněž patří metoda užití takzvaných proxy spojení, jež slouží jako prevence přímého přístupu k hostu (v případě cloudového řešení virtuálnímu stroji). Zaběhnutou praxí je rovněž užití bezpečných protokolů, například GRE tunelů v kombinaci s výměnou klíčů.

Zcela obvyklé je i využití bezpečných aplikačních protokolů SSH/sFTP, HTTPS SSL-TLS. Data přenášená těmito protokoly lze v případě nutnosti ještě dále šifrovat vhodnými kryptografickými šiframi.

2.2.5 Bezpečnost hosta

Tato vrstva se zaměřuje na hosta. V případě cloudu je ním pak myšlen server či počítač, který obsahuje uživatelská data (aplikaci, službu, informace). Host jako takový je tedy kombinace hardware a operačního systému umístěná v síti (cloudu). Opatření na této vrstvě se tedy vztahují k ošetření slabin operačního systému, zabezpečení hardwaru samotného a umístění hosta v rámci síťové topologie.

Bezpečnost na této vrstvě se zajišťuje vhodným nastavením privilegií pro jednotlivé uživatele, či skupiny uživatelů systému v závislosti na typu použitého operačního systému. Těmto požadavkům by měla vyhovovat i samotná aplikace běžící na daném serveru. V souvislosti s operačním systémem je třeba na této vrstvě řešit i problematiku vhodnosti použitého typu autentizace a jejich vzájemné kompatibility.

Na této úrovni je také řešeno šifrování dat a souborů uložených na serveru. S tím je úzce spojeno logování a auditování prováděných operací. Pomocí těchto auditů a logů je pak možné efektivně dohledávat potencionální útoky.

Zabezpečení hardwaru se pak realizuje uzamčením, vypnutím nevyužívaných IP portů či zavedením firewallu pro omezení přístupu. Zásadní je pak včasná aplikace aktuálních záplat a updatů pro daný operační systém.

Testování zabezpečení této vrstvy se pak obvykle provádí pomocí komerčně dostupných nástrojů schopných vyhledávat slabiny a simulovat známé typy útoků.

2.2.6 Bezpečnost aplikace

Tato úroveň aplikace se zaměřuje na životní cyklus aplikací, které jsou přímo přístupné z perspektivy koncového uživatele. Vhodným příkladem mohou být veškeré webové aplikace. Jednotlivými fázemi, které je nutno uvažovat jsou fáze designu, vývoje, nasazení, vylepšení a údržby.

K dosažení bezpečnosti na této úrovni je opět využívána autentizace, přístupová pravidla, šifrování dat používaných aplikací a využívání dedikovaných rozhraní (portů). Rovněž je třeba dbát na včasné aplikování bezpečnostních záplat, pravidelnou revizi užitého kódu aplikace, předpovědí možných rizik a hrozeb. Podobně jako u předchozí vrstvy i zde by mělo být přítomno auditování a logování událostí pro potřeby investigace potencionálních útoků na aplikaci.

2.2.7 Bezpečnost dat

Bezpečnost dat, jak už název napovídá, se zabývá zabezpečením dat před neoprávněným přístupem nebo poškozením dat uložených na serverech, databázích a webových stránkách. V praxi se bezpečnost dat často označuje jako bezpečnost informací nebo bezpečnost počítačů. Součástí této vrstvy je i ochrana vůči poškození dat (korupci).

Metodami pro zajištění bezpečnosti této vrstvy jsou šifrování disků (Luks), souborů, kontrola přístupu, zápisu a čtení dat [18]. Strategie pro zálohování a obnovu dat je možné rovněž řadit pod tuto úroveň. Využívá se softwarových i hardwarových metod, přičemž zálohovaná data musejí být uloženy na zabezpečeném místě (sejf). V případě telekomunikační aplikace je povinnost uchovávat informace daná platnou legislativou.

3 ANALÝZA RIZIK A BĚŽNÉ ÚTOKY NA CLOUD

Jak, již bylo řečeno, cloudový model výpočtu je poměrně mladý, avšak soudobý trend jasně ukazuje narůstající zájem firem o cloudově distribuované technologie. S tímto zájmem lze rovněž očekávat nárůst počtu útoků vedených proti takovýmto systémům. V následujícím textu bude popsán několik možných pohledů na bezpečnostní rizika spojená s cloudovými řešeními a jednotlivé typy nejčastějších útoků cílených na služby provozované v cloudu.

3.1 Analýza rizik

V následujícím textu je popsána analýza rizik pro typický systém hostovaný v cloudovém prostředí. Pohledy na bezpečnostní rizika zahrnují pohled organizační a technický. V praxi je obvykle součástí analýzy rizik i část, věnující se legislativním rizikům spojeným s provozem cloudového systému, avšak tyto v práci nebudou zahrnuty. V závěru této sekce uvádím pro úplnost seznam rizik, jež nejsou přímo specifické pro cloudovou distribuci, ale je nutné je zahrnout pro úplnost analýzy.

3.1.1 Organizační rizika

LOCK-IN

Lock-in je označení pro situaci, kdy z důvodu nestandardního řešení (formát dat, nástroje virtualizace, procedury) není zaručena přenositelnost cloudového řešení. V případě, že se zákazník rozhodne přejít k jinému poskytovateli, migrace takovéhoto řešení je velmi náročná. Toto riziko někdy bývá ještě umocněno snahou poskytovatele (přímo či nepřímo) zamezit portabilitě zákazníkových služeb a dat. Podobný efekt může mít i akvizice či zánik poskytovatele. Lock-in se může vztahovat buďto na API vrstvu (aplikační programovací rozhraní umožňující komunikaci s cloudovým systémem a jeho správu) nebo na hardwarovou vrstvu (specializované komponenty).

Pro potřeby migrace virtuálních strojů a jejího usnadnění se zavedl standardizovaný formát OVF [19]. Hlavním rizikem je v případě lock-in problému případná ztráta dat v důsledku jejich nepřenositelnosti.

Ztráta kontroly

V případě cloudových systémů se uživatel nevyhne předání části zodpovědnosti za bezpečnost systému cloudovému poskytovateli. Mohou tak nastat konflikty mezi zákazníkovým zabezpečením a cloudovým prostředím (software hledající a testující bezpečnostní chyby, skenování portů a další). Na druhé straně SLA nemusí obsahovat závazek k danému typu zabezpečení, tudíž ponechává potencionální trhlinu v zabezpečení systému [20].

Dodržení závazků

Jak již bylo zmíněno v textu výše, společnosti bývají auditovány a certifikovány důvěryhodnými společnostmi. Pro získání daného certifikátu tak musí zajistit splnění bezpečnostních požadavků těchto institucí. Tento bod analýzy rizik uvažuje schopnost dostát daným závazkům v případě migrace do cloudu, či mezi cloudovými poskytovateli.

Poškození obchodní reputace v důsledku aktivit na sdílených zdrojích

Sdílení výpočetních technologií sebou přináší mnohá rizika. Jedním z nich je případ, kdy škodlivá či kontroverzní činnost jednoho uživatele, může negativně ovlivnit obchodní pověst jiného uživatele. Příkladem může být spamování, skenování portů a tak dále.

Zánik cloudového prostředí nebo jeho selhání

Závažným rizikem pro cloudové systémy je zánik, či omezení nabízených služeb cloudového poskytovatele. Tato situace může být důsledkem nevhodné obchodní strategie poskytovatele, nedostatečného finančního zázemí či důsledkem konkurenčního nátlaku. Výsledkem může být lock-in popsaný v textu výše. Řešením je migrace k jinému poskytovateli. Jako prevence slouží užití standardizovaných nástrojů, formátů a hardwaru.

Akvizice poskytovatele cloudu

Akvizice poskytovatele cloudu může v konečném důsledku znamenat změnu v bezpečnostní politice poskytovatele a tudíž způsobit porušení v dodržení závazků.

Výpadek na straně dodavatele

V některých případech se poskytovatelé cloudových řešení uchylují k využívání prostředků třetích stran. V takovém případě je opět nutné dodržovat zásady bezpečné komunikace, přičemž celková míra zabezpečení závisí na každém jednotlivém prvku řešení včetně těchto třetích stran. V případě výpadku či ztráty této třetí strany tak nastává riziko ztráty dostupnosti, důvěrnosti či integrity. Zásadními prvky při takovéto distribuci služeb jsou jasně definované sféry zodpovědnosti zúčastněných stran a jejich vzájemná koordinace.

3.1.2 Technická rizika

Vyčerpání výpočetní kapacity

Cloudové služby patří mezi takzvané on-demand služby (služba na vyžádání). Z tohoto důvodu zde existuje riziko úplného vyčerpání výpočetní kapacity cloudového řešení. Jak již bylo řečeno, mnohdy se množství alokovaných výpočetních zdrojů upravuje v závislosti na aktuální míře požadavků, tedy podle statistických požadavků. Bohužel, algoritmy generující tyto statistiky bývají náchylné k nepřesnostem. Chyba ve výpočtu tak může způsobit buďto nedostatek alokovaných výpočetních zdrojů, anebo naopak neadekvátní náklady za přebytečné množství alokovaných zdrojů. Výsledkem může být nedostupnost služby na jedné straně či ztráta profitability provozované služby na straně druhé. Toto vyčerpání může být důsledkem takzvaného DDoS útoku, který je popsán v následující kapitole.

Selhání izolace

Multitenantnost a sdílení zdrojů jsou dvě hlavní definující charakteristiky cloudových řešení. Výpočetní kapacita, úložiště a síť jsou sdíleny mezi množstvím uživatelů.

Tato třída rizik zahrnuje selhání bezpečnostních mechanizmů zajišťujících oddělení úložišť, paměti a směrování mezi jednotlivými uživateli (útoky pomocí injektování SQL dotazy, útoky na postranní kanály). Následkem útoků využívajících tohoto rizika může být ztráta cenných či citlivých dat a nedostupnost služby koncovým uživatelům.

Zneužití zevnitř – Zneužití privilegované funkce

Škodlivá činnost člověka uvnitř organizace (poskytovatel nebo zákazník) s privilegovaným přístupem ke cloudovému vybavení (administrátor, auditor) může mít potencionální dopad na důvěrnost, integritu a dostupnost dat, služby, síťový provoz a nepřímo tak ovlivnit reputaci organizace a důvěru uživatelů.

Kompromitace portálu pro správu cloudového prostředí

Většina veřejných cloudových poskytovatelů provozuje webová rozhraní, která umožňují jeho zákazníkům spravovat své cloudové prostředí. V případě, že je toto rozhraní kompromitováno, útočník je schopen provádět veškeré akce, které toto rozhraní umožňuje (manipulace s virtuálními servery, přístup ke službě). Tato potencionální hrozba může být zmírněna přijetím dalších bezpečnostních prvků (tokeny, ověřování pomocí SMS a další).

Zachytávání dat během přenosu

Jelikož je cloud computing distribuovaný typ architektury, vyžaduje ve srovnání s tradičními modely větší množství datových přenosů. Příkladem budiž synchronizační data jednotlivých virtuálních strojů a výměna informací mezi fyzickými stroji hostující danou službu a jejími uživateli. Tato třída rizik tak zahrnuje nejrůznější varianty útoků typu sniffing, spoofing, man-in-the-middle [21]. Nejčastěji užitým ochranným prostředkem vůči těmto rizikům jsou VPN tunely.

Nedostatečné vymazání dat

V případě změny cloudového poskytovatele, je potřeba zajistit korektní smazání dat v úložišti cloudového poskytovatele. Důvodem je potřeba dostát bezpečnostním požadavkům plynoucích z certifikačních závazků a SLA. Z důvodu sdílení výpočetních zdrojů, zejména úložiště s jinými zákazníky, je proto potřeba uvážit možné komplikace spojené s korektním smazáním dat. Vhodné šifrování dat snižuje míru tohoto rizika.

Distribuované zahlcení služby (DDoS)

Podrobněji bude popsáno v následující kapitole.

Ztráta bezpečnosti šifrovacích klíčů

Tato třída rizik zahrnuje odhalení tajných klíčů (SSL, šifrování dat, privátní klíče zákazníků) nebo hesel třetí stranou, jejich poškození a ztrátu. Rovněž zde patří i rizika spojená s jejich neoprávněným užíváním pro potřeby autentizace a nepopiratelnosti (digitální podpis).

Riziko skenování a sondování

Sondování a skenování prostředí může být součástí počáteční fáze útoku vedeného vůči cloudovému prostředí. Využívá se k získání informací o cloudovém prostředí, bezpečnostních mechanizmech a k nalezení slabin systému.

Zneužití hypervisoru

Jak již bylo uvedeno dříve v textu, cloudová architektura je obvykle spravována pomocí hypervisoru, který kontroluje jeho jednotlivé fyzické součásti. V případě jeho kompromitace by tak útočník získal úplnou kontrolu nad cloudovým prostředím. Stejně jako každý software, tak i vrstva, v níž se nachází hypervisor může obsahovat slabiny a je tak možné cílit útok vůči ní. Útočník může kompromitovat hypervisor pomocí útoku vedeného z virtuálního stroje či API.

3.1.3 Hrozby nespecifické cloudu

V předchozím textu byly zevrubně popsány rizika a hrozby specifické pro cloudová řešení. Aby byla tato analýza kompletní, je nutné uvést ještě i rizika, jež nejsou specifická pro cloudová řešení, ale přesto představují potencionální hrozbu pro bezpečnost jakéhokoli cloudového systému.

- výpadek komunikační sítě,
- správa sítě (zahlcení sítě, výpadky spojení, neoptimální využití sítě),
- změna síťového provozu,
- privilegovaná eskalace,
- útoky prostřednictvím sociálního inženýrství (imitace jiné osoby),
- ztráta či kompromitace operačních záznamů,
- ztráta či kompromitace bezpečnostních záznamů,
- ztráta nebo krádež zálohovaných dat,
- neautorizovaný přístup k vybavení (zahrnuje fyzický přístup k hardwarovým strojům a ostatnímu vybavení),
- krádež výpočetního vybavení,
- přírodní katastrofy.

Obecně platí, že rizika související s následky přírodních katastrof v případě cloudových architektur znamenají pro provozovatele nižší míru rizika, než je tomu u ostatních tradičních datových modelů. Za tento fakt lze vděčit zejména síťové a fyzické redundanci, kterou obvykle poskytovatelé cloudu nabízejí.

3.2 Přehled nejčastějších útoků

3.2.1 Odmítnutí služby (DoS)

Odmítnutí služby je technika útoku, kdy se útočník pokouší narušit dostupnost služby zasíláním nadměrného množství požadavků na danou službu [22]. Vzhledem k vysokému počtu uživatelů sdílejících cloudové výpočetní prostředky lze tvrdit, že cloudová architektura je výrazně zranitelnější vůči DoS útokům v porovnání s jinými typy výpočetních modelů. V případě útoku DoS totiž není poškozen či omezen jeden uživatel, nýbrž celá skupina uživatelů. Rovněž je třeba uvažovat specifické chování cloudového řešení při nárůstu vytížení systému. Jak už bylo zmíněno v předchozím textu, v případě nedostatku výpočetní kapacity dochází k dynamickému navyšování přidělených výpočetních zdrojů. Systém tak alokuje pro výpočet další zdroje (virtuální stroje, instance služby, paměť). Díky této vlastnosti tak cloudově řešený systém není limitován hardwarovými omezeními a neexistuje tak zde horní hranice vyčerpatelnosti systému.

V podstatě to pak vypadá, že cloud pracuje proti útočníkovi (snaží se kompenzovat množství útočníkových žádostí navyšováním výpočetní kapacity), ale ve skutečnosti narušiteli pomáhá, protože mu umožňuje omezit celkovou dostupnost systému. Útočník navíc nemusí útočit na veškeré servery poskytujících danou službu, ale stačí mu se zaměřit na jedinou cloudovou adresu, aby dosáhl celkového výpadku dané služby [23]. Dalším rizikem pro provozovatele služby v cloudu napadené tímto typem útoku pak je samotné navýšení přidělených výpočetních zdrojů a s tím související poplatky poskytovateli cloudu.

3.2.2 Injekce malwaru do cloudu

Oproti předchozímu útoku se tato technika zaměřuje na snahu umístit škodlivý software či napadený virtuální server do cloudového systému. Takovýto malware, zde pak působí škodu provozovateli systému ve prospěch útočníka. Spektrum možných cílů je široké, počínaje odposlechem, kompletní změnou funkcionality systému, blokací či omezení systému včetně přístupu k privátním datům uživatelů. Tento typ útoku vyžaduje, aby narušitel vytvořil a umístil do cloudu vlastní škodlivou aplikaci (SaaS nebo PaaS), či virtuální stroj (IaaS). Aby byl takovýto útok úspěšný, narušitel navíc musí zajistit, aby se jeho škodlivý software tvářil jako součást původního systému a nebyl tak odhalen. Pokud škodlivý software splňuje tyto podmínky, požadavky oprávněných uživatelů jsou přesměrovány na útočníkovu zmodifikovanou verzi implementace služby, přičemž je proveden jeho škodlivý kód.

Jakožto opatření vůči tomuto typu útoku se v praxi osvědčila technika ověření integrity dané služby před jejím užitím. V praxi to pak znamená, že v případě vyžádání služby uživatelem, je tato ověřena a až v případě kladného vyhodnocení je umožněno její vykonání. Obvykle se toto realizuje pomocí porovnání hashované hodnoty, která je uložena v image původní služby, s hashovanou hodnotu všech nových instancí služby. Aby mohl útočník umístit svůj škodlivý software do cloudového systému a ten zároveň nebyl odhalen, musel by být narušitel schopen simulovat správnou hodnotu originálního hashe.

Jak již bylo řečeno, hlavním cílem injekce malwaru do cloudu je umístit zmanipulovanou instanci dané služby oběti do cloudu tak, aby část požadavků na tuto

službu byla zpracována touto škodlivou instancí. Aby bylo možné tohoto dosáhnout, útočník musí získat přístup a kontrolu nad daty umístěnými v cloudovém prostředí. Tento útok je hlavním představitelem skupiny útoků služba-cloud [24].

3.2.3 Útoky postranními kanály

Následující typ útoku vůči cloudovým řešením se do jisté míry podobá útokům popsaným v předchozím odstavci. Založen je na získání informace o šifrovacím systému na základě jeho fyzické implementace a jejich postranních kanálů (fyzické projevy probíhajících operací). Těmito zneužitelnými postranními kanály mohou být například znalost taktovací frekvence (časování), energetická spotřeba, elektromagnetické úniky, či zvuk. Tyto znalosti pak může útočník zneužít k útoku na systém. Některé útoky postranními kanály vyžadují hlubokou technickou znalost interní funkcionality systému na kterém je šifrovací systém implementován, zatímco jiné je možné použít bez znalosti vnitřní architektury. V případě cloudu tak podobně jako v případě injekce malwaru, se útočník pokouší umístit do cloudu vlastní stroj či software v blízkosti cílového serveru oběti, odkud posléze sleduje postranní kanály cíleného serveru. Útoky postranními kanály se v praxi projevily jako efektivní způsob zneužití bezpečnostních slabin, jež jsou zaměřené na implementaci šifrovacích algoritmů.

Z tohoto důvodu se dnes věnuje veliká pozornost analýze užitého šifrovacího systému zabezpečení cloudu a jeho jednotlivých součástí. Na základě této analýzy pak obvykle vzniká návrh šifrovacího systému, jehož úkolem je zajistit odolnost vůči těmto útokům [25].

3.2.4 Útoky vůči autentizaci

Praktické užívání cloudových řešení v posledních letech ukázalo, že navzdory zlepšování složitosti a náročnosti autentizačních procesů, autentizace je stále slabinou hostovaných a virtuálních služeb cloudu a stává se tak častým cílem útoků. Existuje široké spektrum způsobů, jakými lze autentizovat uživatele. Velmi často je pak autentizace prováděna na základě něčeho, co daný uživatel zná, něčím co daný uživatel má, nebo něčím čím daný uživatel je [26]. Konkrétním cílem tohoto typu útoků je pak samotný mechanizmus autentizace, který se snaží útočník oklamat.

Pokud bychom uvažovali nad odolností jednotlivých cloudových řešení, pak v současnosti pouze cloudová architektura typu IaaS nabízí potřebnou ochranu (ochrana dat a šifrování) vůči tomuto typu napadení. Tento typ cloudu oproti ostatním nabízí uživateli potřebné prvky zabezpečení a možnosti jejich správy nutné pro zajištění bezpečné komunikace. Výhodou je rovněž odlišnost procesu autorizace nad daty. Ve srovnání s PaaS a SaaS se tedy neprovádí autorizace nad datovými procesy a jejich správou (data patří společnosti, ale jsou hostovány na hardwaru u cloudového poskytovatele) vůči poskytovateli cloudové architektury, ale vůči uživateli, kterým je opět firma provozující službu.

V současnosti většina služeb přístupných z internetu využívá jednoduchou autentizaci pomocí znalosti, obvykle realizovanou pomocí kombinace uživatelského jména a hesla. Výjimkou bývá zabezpečení finančních portálů a internet-bankingu, kde je povětšinou užito různých forem sekundární autentizace. Těmito mohou být například stránkové klíče (ochrana vůči phishingu), virtuální klávesnice (ochrana proti zaznamenání úderů do klávesnice), sdílené tajné otázky a další.

3.2.5 Man-In-The-Middle útok

Jedná se o variantu útoku, kdy se útočník stává aktivním prostředníkem mezi dvěma uživateli a snaží se zachytit a následně číst či měnit probíhající komunikaci. V případě úspěšného provedení tohoto typu útoku lze takovéto cloudové prostředí pokládat za zkompromitované a jeho bezpečnost již nadále není zaručena. Podmínkou pro provedení man-in-the-middle útoku v cloudu je získání přístupu do tohoto cloudu. Tento typ útoku se v cloudovém prostředí obvykle objevuje jako doprovodný jev výše popsaných útoků v případě, že se útočníkovi podařilo získat neoprávněný přístup do cloudu. Následně se útočník zaměřuje na slabiny komunikace mezi jednotlivými účastníky (šifrování).

4 METODY PŘENOSU DAT MEZI UŽIVATELEM A CLOUDEM

V této kapitole se v práci věnuji nejčastějším metodám přenosu dat mezi uživatelem a aplikací či virtuálním strojem v cloudovém prostředí. V současné době se primárně využívá pro bezpečné připojení ke cloudovým řešením VPN tunelů (virtual private network). Ty nabízejí uživatelům bezpečné šifrované spojení typu site-to-site nebo remote-access [27]. Existuje množství různých VPN klientů založených na množství odlišných technologií (openVPN, SSTP, L2TP, IPSec a další). Jelikož je velké množství VPN klientů založeno na technologiích IPSec a SSL/TLS, v dalším textu budou tyto popsány podrobněji.

4.1 IPSec

IPSec (Internet Protocol Security) je průmyslový standard, který je určen k zajištění bezpečnosti informací v komunikačních sítích založených na přenosovém protokolu IP. Zajišťuje autentičnost přenášených paketů a důvěrnost přenášených informací [28]. Součástí protokolu IPSec jsou dílčí protokoly, konkrétně pak protokol AH (Authentication Header) pro zajištění autentičnosti paketu a protokol ESP (Encapsulating Secutrity Payload) pro zajištění důvěrnosti a autentičnosti přenášených dat.

Protokol AH tak zodpovídá za autentizaci odesílatele a příjemce a integritu dat obsažených v hlavičce paketu. Nevýhodou samotného protokolu AH je, že nešifruje vlastní přenášená data. Z tohoto důvodu je používán v kombinaci s protokolem ESP. Ten zajišťuje šifrování přenášených dat a tak vhodně doplňuje protokol AH. Pro oba tyto protokoly a jejich kombinace existují dvě varianty provozu v závislosti na druhu probíhající komunikace.

První z nich je varianta transportního módu, která je obvykle využívána pro přenos mezi koncovými zařízeními, jež disponují veřejnou IP adresou. Druhá varianta, takzvaná tunelová, se využívá pro bezpečný přenos mezi branami různých sítí.

V obou případech platí, že koncovým zařízením musí být známy potřebné šifrovací údaje. Obvykle jimi bývá použitá hašovací funkce, šifrovací algoritmus, provozní mód šifrátoru a klíče. Výměna těchto šifrovacích parametrů obvykle probíhá pomocí asymetrické kryptografie (protokol IKE). Tyto sjednané parametry přenosu jsou pak uchovávány v databázi SA (Security Association).

Autentizace IPSec probíhá pomocí MAC, konkrétně její varianty HMAC (Keyed-Hashing for Message Authentication). Jedná se tedy o hašovací funkci, v současnosti realizovanou algoritmy SHA-2, SHA-3 nebo BLAKE [29].

Samotné šifrování přenášených dat je pak realizováno různými šiframi. V praxi bývají nejčastěji používány algoritmy AES, TripleDES.

4.2 SSL

Podobně jako IPSec, tak i SSL („Secure Socket Layer“) je průmyslový standard bezpečného přenosu v komunikačních sítích (v současnosti již ve verzi TLS) [30]. Oproti protokolu IPSec pracujícím na IP vrstvě, protokol SSL pracuje na vrstvě transportní, tedy zajišťuje bezpečný přenos pomocí protokolu TCP. Zodpovídá jak za autentizaci komunikujících stran, tak i za autentičnost a důvěrnost přenášených informací.

Princip SSL je následující. Zpráva je nejprve fragmentována na segmenty a ty jsou vhodně komprimovány. Poté je k těmto segmentům připojen autentizační kód vypočítaný pomocí „hašovací“ funkce. Takto vzniklý blok je následně šifrován a protokol TCP pak zajistí spolehlivý přenos dat k příjemci, kde jsou provedeny inverzní operace. Příchozí bloky jsou dešifrovány a následuje kontrola autentizačních kódů jednotlivých segmentů. V případě kladného výsledku je provedena dekomprimace segmentů a je sestavena původní zpráva. Pro účely hašování se podobně jako u technologie IPSec využívá algoritmů SHA-2, SHA-3 nebo BLAKE [29].

V případě metody SSL se k šifrování přenášených dat v praxi používají symetrické šifry, přičemž nejpoužívanějšími šiframi jsou pak 3DES, AES.

Podobně jako u IPSec technologie, tak i v případě SSL je nutné před samotným přenosem ustavit tajné hodnoty s kterými pak pracují bezpečnostní a šifrovací algoritmy. Těmito obvykle jsou dvojice autentizačních klíčů, inicializačních vektorů a šifrovacích klíčů, vždy po jednom pro odesílatele a jednom pro příjemce. Samotný proces výměny těchto dat je realizován protokolem SSL Handshake, zatímco přenos samotný a šifrování s ním spojené je vykonáváno protokolem SSL Record. Současně SSL Handshake provádí autentizaci komunikujících stran pomocí asymetrické kryptografie – ověření platnosti certifikátu.

SSL Ověřování certifikátů

Ověřování certifikátů je jedním ze základních charakteristik technologie SSL. V současnosti se jedná o standardní metodu bezpečného přístupu k webovým stránkám obsahující citlivé, či osobní informace a v internetovém bankovnictví. Standardní SSL certifikát tak obsahuje doménové jméno, název serveru či jméno hosta v kombinaci s organizační identitou a lokací. Nyní bude popsán zjednodušený princip ověřování certifikátu druhé strany předcházející samotnému šifrovanému přenosu.

Uživatel si vyžádá certifikát druhé strany (webový server v cloudu). Ten musí obsahovat veřejný klíč druhé strany a současně by měl být podepsán privátním klíčem (digitální podpis) důvěryhodné certifikační autority. Uživatel rovněž disponuje veřejným klíčem certifikační autority (popř. o něj může zažádat) a pomocí něj ověří, zda certifikát zaslaný druhou stranou byl opravdu podepsán certifikační autoritou.

Následně uživatel vygeneruje symetrický klíč pro potřeby přenosu a zašle jej druhé straně zašifrovaný pomocí veřejného klíče druhé strany. Ta jej dešifruje pomocí svého privátního klíče a komunikace nadále probíhá pomocí rychlejší symetrické šifry a takto ustanoveného klíče.

5 NÁVRH DATOVĚ ÚSPORNÉHO ZABEZPEČENÍ PŘENOSU DAT MEZI UŽIVATELEM A CLOUDEM

V následujícím textu se budu zabývat samotným návrhem zabezpečení cloudového řešení. Popsány jsou tak technologie a postupy zvolené pro samotnou implementaci cloudového zabezpečení, včetně užitého softwaru a důvodů výběru daných technologií a prostředků zabezpečení.

Prvním nutným krokem předcházejícím samotnému návrhu zabezpečení je volba typu cloudového prostředí, jeho typu a použité technologie. Pro potřeby této práce jsem zvolil cloudovou architekturu odpovídající typu IaaS, která byla popsána v textu výše. Tato distribuce nabízí potřebnou míru spravovatelnosti cloudové infrastruktury a použitých technologií.

Dalším krokem je výběr konkrétní varianty hostingu cloudu. Existuje poměrně velké množství různých realizací samotného cloudového prostředí, počínaje hostingem u komerčního cloudového poskytovatele, simulací cloudového prostředí virtualizačními nástroji a vytvořením vlastního cloudového prostředí konče.

Vytvoření vlastního cloudového prostředí se dá pokládat za jednoznačně nejvěrnější možnost simulace reálného cloudového prostředí a nabízí praxi nejlépe odpovídající možnosti aplikace bezpečnostních opatření. V neprospěch této varianty však silně hovoří její celková složitost a finanční náročnost.

Jednodušší a snadnější možností je použití veřejně dostupného multiplatformního nástroje pro virtualizaci např. VirtualBox, VMware Player, VMware Workstation [31]. Jelikož mám osobní zkušenost s distribucí VirtualBox, právě tato varianta se jeví jako nejvhodnější volba. Za její nevýhodu však pokládám absenci prvků typických cloudovému řešení (síťovou topologii, hardware na pozadí virtuálních strojů). Z těchto důvodů jsem se rozhodl pro možnost třetí, hosting virtuálních strojů u cloudového poskytovatele.

Podobně jako v případě vlastního cloudu, i v tomto řešení bylo nutné uvážit náklady spojené s hostováním cloudu. Díky mé současné pozici v cloudovém týmu mého zaměstnavatele se mi však otevřela možnost umístit své řešení (zabezpečované servery) do cloudového prostředí profesionálního poskytovatele bez nutnosti odvádět poplatky za užité výpočetní prostředky (testovací provoz). Takto umístěné řešení mi umožní práci s reálným vybavením (firewally, směrovače), přičemž budoucí zabezpečované servery budou hostované na reálném hardwaru. Vybraný poskytovatel nabízí pro potřeby síťové zprávy softwarový směrovač, firewall a load-balancer společnosti Fortinet, pro nějž se v praxi užívá název INA (Internet Network Appliance). Cloudové řešení tohoto providera je pak postaveno na dobře známé technologii VMware.

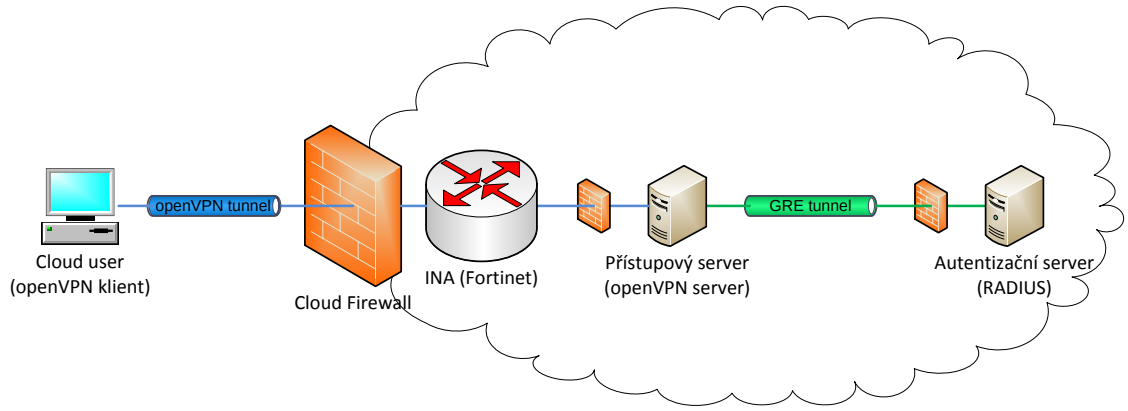
Dalším důležitým prvkem ovlivňujícím výslednou podobu mého řešení byl výběr použitého operačního systému zabezpečovaných serverů (virtuálních strojů). Po pečlivém uvážení jsem se rozhodl pro operační systém na bázi Linuxu. K této volbě jsem dospěl na základě uvážení výhod spjatými s linuxovými distribucemi. Nespornou výhodou

linuxových systémů je zejména jejich dostupnost (open-source), softwarová vybavenost a jejich škálovatelnost. Další neopomenutelnou výhodu spatřuji v jednoduchosti vytváření skriptů, kterými lze ovlivňovat chování sytému. Jako takový jsem se rozhodl pro operační systém CentOS ve verzi 7. Rozhodl jsem se pro něj z důvodu podobnosti s průmyslovým Red Hat Enterprise Linuxem společnosti RedHat, který však nenabízí bezplatnou verzi tohoto operačního systému.

Jako řešení jsem se rozhodl nainstalovat do cloudového prostředí dvojici serverů, na nichž bude nahrán zmíněný operační systém. První ze serverů bude vystupovat jako takzvaný přístupový server a bude na něm terminován tunel VPN. Tento tunel VPN jsem se rozhodl realizovat pomocí veřejně dostupného řešení – openVPN. Tato varianta bude tedy vyžadovat instalaci a konfiguraci OpenVPN serveru na straně přístupového serveru a klienta na straně připojované stanice. Alternativní možností k OpenVPN by bylo vytvoření vlastního VPN klienta a serveru. Implementace tohoto klienta a serveru by byla provedena v programovacím jazyku C++, který nabízí široké množství funkcí a knihoven pro bezpečnou síťovou komunikaci.

Druhý ze serverů bude sloužit jako autentizační server, přičemž se bude využívat autentizační služby RADIUS. Tato služba bývá realizována pomocí serveru, na který jsou zasílány autentizační požadavky připojených serverů. Tento server bude rovněž sloužit jako úložiště a cíl pro přenos testovacích dat, pomocí kterých plánuji simulovat a měřit parametry přenosu v závislosti na použitých ochranných prvcích.

Aby byla komunikace mezi těmito servery chráněna proti útokům typu man-in-the-middle a současně izolována od zbytku prostředí, hodlám jej realizovat pomocí zabezpečeného šifrovaného GRE tunelu. Toto řešení dovoluje serverům překládat své privátní IP adresy na adresy používané tunelem. Kompletní architekturu podrobně vykresluje následující obrázek.



Obr. 3: Model návrhu připojení uživatele k serveru.

Současně budou oba servery podléhat standardním bezpečnostním zabezpečením. Mezi tyto standardní opatření patří lokální firewally, uživatelské účty a příslušná oprávnění a komunikace bezpečným protokolem SSH. Servery budou nadále chráněny primárním cloudovým firewallem (Fortinet) provozovaným cloudovým poskytovatelem.

Datové úspory bych chtěl dosáhnout pomocí datové redukce. Pojem datová redukce zahrnuje techniky deduplikace a komprese dat. Kompresí dat je myšlena vhodná komprimace za účelem snížení celkového objemu dat uchovávaného na disku.

Deduplikaci dat lze charakterizovat jako specializovanou kompresi dat, pomocí které jsou eliminovány duplikátní kopie dat v úložišti. Úložiště, jež používají tyto techniky, jsou odborně označovány jako takzvané single-instance úložiště. V návazné práci bych chtěl tyto techniky užít prostřednictvím veřejně dostupného single-instance úložiště OpenDedup.

6 REALIZACE – PLATFORMA

Prvním krokem v realizaci samotného návrhu cloudového úložiště bylo vybrat vhodného cloudového providera a v souvislosti s tím také odpovídající virtualizační nástroj. Tato kapitola je věnována popisu zvolené platformy, virtualizačního nástroje a použitého operačního systému. Jsou zde uvedeny a vysvětleny důvody, kvůli kterým byli dané varianty realizace upřednostněny před jinými a použity ve výsledné implementaci návrhu cloudového úložiště. Tato kapitola se rovněž zaobírá stručným popisem zvoleného operačního systému.

6.1 Výběr poskytovatele cloudového prostředí

Jak již bylo zmíněno v samotném návrhu realizace cloudového řešení, díky mému současnému zaměstnání bylo možné využít služeb reálného komerčního poskytovatele cloudových řešení. Potencionálními kandidáty tak byl AWS Amazon, Navisite a NTT. Při výběru poskytovatele cloudového prostředí byly zkoumány a porovnávány tyto hlavní kritéria: dostupnost, jednoduchost, cena. Následující text se tak zabývá jednotlivými poskytovateli, jejich výhodami a nevýhodami. V samotném závěru je pak uveden vybraný cloudový provider a vysvětleny důvody proč byl daný poskytovatel vybrán.

Jako první jsem se zabýval největším z možných poskytovatelů Amazon Web Services (dále jen AWS) a konkrétně pak jejich cloudovým řešením Elastic Compute Cloud (EC2) [32]. Jedná se o bezkonkurečně největšího a na trhu nejúspěšnějšího poskytovatele cloudových řešení s bohatou zkušeností a disponuje tak obrovskou technologickou základnou. Jednoznačnou výhodou tohoto poskytovatele tak je dostupnost, stabilita a vysoká kvalita technické podpory a dokumentace. Z této silné pozice na trhu pak vyplývá jistá neochota AWS poskytovat kustomizace pro zákazníky (nedostupnost API, testovací stroje). Další nevýhodou se pak ukázala velmi limitovaná možnost správy síťových prvků. Posledním faktorem hovořícím proti této variantě byla fixní cena instancí (virtuální stroje). Tyto důvody vedly k rozhodnutí využít služeb menšího cloudového poskytovatele.

Další možností volby poskytovatele bylo cloudové řešení společnosti Navisite [33]. Jednoznačnou výhodou této možnosti je dobrá znalost prostředí a předchozí zkušenost s prací v tomto prostředí. Další výhodou je plný přístup k síťovým prvkům a flexibilita poskytovatele (dostupnost bezplatného testovacího prostředí). Nevýhodou je pak nedostupnost API a nutnost využití VPN klienta firmy Cisco – AnyConnect a takzvaného airlocku, stavu, kdy je virtuální stroj izolován od veškeré okolní infrastruktury a je přístupný pouze pomocí VPN spojení. Vzhledem k zaměření této práce (síťová konfigurace a bezpečnost) bylo možné očekávat častou rekonfiguraci vyžadující restart konfigurovaných serverů. V případě omylu by tak oprava vedla k nutnosti zdlouhavého přepínání mezi normálním stavem a režimem airlocku.

Kvůli nastíněným problémům jsem se nakonec uchýlil k řešení realizovaném v distribuci poskytované společností NTT [34]. NTT web portál umožňuje plnou kontrolu síťových prvků, správu virtuálního stroje pomocí kustomizovaného API, cenově výhodnou instalaci (trialové prostředí) včetně užití vlastních OVA/OVF „imagů“ pro instalaci OS. Oproti ostatním porovnávaným, tento poskytovatel nevyžaduje užití

softwaru třetí strany ani instalaci dalších certifikátů (související potencionální bezpečnostní riziko je v této práci ošetřeno) pro přístup k serveru.

6.2 NTT

Tento odstavec popisuje Enterprise Cloud distribuci poskytovanou firmou NTT. V současné době NTT disponuje datacentry po celém světě, v nichž jsou uchovány hardwarové servery. Tyto HW stroje představují výpočetní zdroje poskytované cloudovým providerem pomocí virtualizačních nástrojů cloudovým zákazníkům. Tento Enterprise Cloud se nadále dělí na menší podcelky nazývané VR (virtual room). V jednotlivých virtuálních místnostech jsou pak uloženy samotné servery. Každý z těchto serverů náleží takzvanému resource poolu, skupině alokovaných výpočetních zdrojů.

Virtuální stroje v cloudu je možné spravovat pomocí webového uživatelského rozhraní, či pomocí příkazů API z příkazové řádky. Těmito způsoby uživatel komunikuje přímo s hypervisorem, v tomto případě VMware vCloud Directorem. Kvůli rychlosti prováděných operací jsem zvolil instalaci pomocí API příkazů. Síťovou konfiguraci, překlad adres NAT, směrování a firewallová pravidla je v současné době možné konfigurovat pouze pomocí webového grafického rozhraní. Webový portál dále disponuje webovou konzolí, pomocí které lze provádět změny konfigurace virtuálního serveru. Virtuální stroje v tomto cloudovém řešení mohou disponovat až 7 různými fyzickými zařízeními NIC (network interface controller).

Portál rovněž nabízí služby monitorování umístěných serverů, zálohování souborů a virtuálních strojů, správu tiketů a úložiště OVA „imagů“ pro instalaci virtuálních strojů.

6.3 vCloud Director, vCentre, vSphere

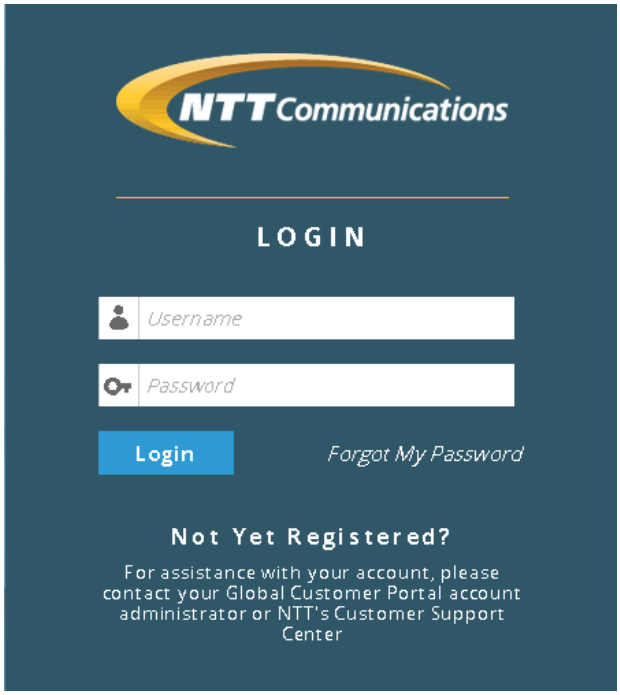
V této podkapitole je blíže popsán použitý virtualizační nástroj vCloud Director [35] vyvíjený firmou VMware a související software nutný pro provozování cloudu. Cloudová řešení postavená na vCloud Directoru jsou založena na principu přidělování výpočetních, síťových a diskových zdrojů existujících datacenter takzvaným virtuálním datacentrům (virtual room) a poskytování jejich služeb uživatelům v podobě webových katalogových služeb (například zmíněný EC). Jedná se o softwarové řešení, které umožňuje společnostem budovat bezpečné, multitenantní privátní cloudová úložiště.

Samotný nástroj vCloud Director poskytuje uživateli abstrahovaný pohled a přístup k hardwarovým výpočetním zdrojům a de facto tak vytváří cloud. VCD se tak nezabývá vytvářením, správou a konfigurací infrastruktury na hardwarové vrstvě. Tuto funkcionalitu zajišťuje jiný, pro cloud nezbytný produkt společnosti VMware – vCenter [36]. Program vCenter umožňuje uživateli vytváření a správu vysoce dostupných distribuovaných klusterů, konfiguraci distribuovaného virtuálního přepínače, přidávání hostů a další. vCloud Director si lze tedy představit jako vyšší vrstvu, jejíž hypervisor komunikuje s vCentrem a zprostředkovává tyto informace v abstrahované formě uživateli.

Obvykle se pak vCloud Director používá ve spojení s dalším nástrojem firmy VMware, programem vSphere. Nástroj vSphere lze považovat za klienta poskytující uživatelům grafické rozhraní umožňující přímou komunikaci s vCloud Directorem.

VMware nabízí jak webovou verzi vSphere klienta, tak i instalovatelného softwarového klienta. Toto spojení dovoluje poskytovatelům nabízet vysoce flexibilní a inovativní prostředí a současně podporuje trend zvyšování bezpečnosti a efektivity provozu cloudových řešení.

Je to právě vCloud Director, který podporuje a poskytuje backend pro webové portály cloudových poskytovatelů. Kromě tohoto portálu, přímého přístupu pomocí vSphere klienta je možné komunikovat s vCloud Directorem rovněž pomocí API. Jedná se o otevřené RESTful API podporující skriptovaný přístup, upload a download virtuálních aplikací a přenositelnost. Ta je zaručena použitím otevřeného virtualizačního formátu OVF.



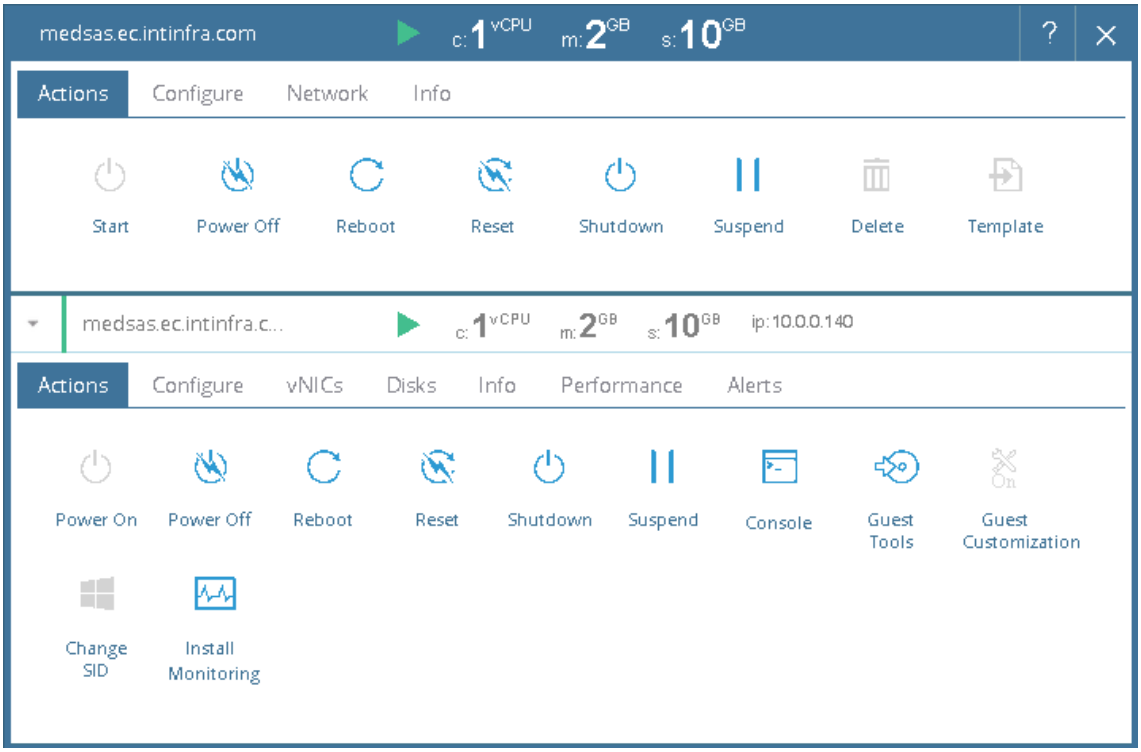
Obr. 4: Ukázka autentizace uživatele na vstupu do cloudového portálu.

6.4 Hardwarové parametry

Následující krok, který bylo nutné učinit, bylo rozhodnutí o velikosti výpočetních zdrojů alokovaných virtuálním strojům navrhovaného řešení. Při rozhodování byl kladen důraz na úspornost řešení a současně na optimální výkon výsledného systému. Z toho důvodu bylo nutné zjistit minimální výpočetní zdroje v souvislosti s použitým operačním systémem a dále pak tyto alokované zdroje přizpůsobit na jednotlivých serverech v závislosti na použitém softwaru a instalovaných aplikacích.

Jak již bylo řečeno v části zabývající se návrhem cloudového úložiště, pro svou implementaci jsem zvolil operační systém CentOS 7 [37]. Proto bylo nutné vyhledat výrobcem doporučené minimální parametry výpočetních zdrojů. Dle informací nalezených na webových stránkách [38] jsem tedy zvolil pro každý server nastavení 1 vCPU s 1 GB paměti RAM a 10 GB diskové paměti. Jelikož jsem v průběhu

implementace návrhu objevil swapování na autentizačním serveru (RADIUS, MySQL), bylo tuto konfiguraci nutné upravit. Pro lepší výkon jsem tedy navýšil paměť RAM na konečných 2 GB.



Obr. 5: Příklad reprezentace virtuálního stroje v NTT.

Z obrázku 5 lze vyčíst velikosti alokovaných výpočetních zdrojů, stav stroje a přiřazenou IP adresu

6.5 Operační systém – CentOS 7

Pro úplnost budou v následujícím textu probrány základní charakteristiky zvoleného operačního systému. Jak již bylo v návrhu zmíněno operační systém jsem volil na základě jeho dostupnosti, bezplatnosti, předchozí zkušenosti a zpětné kompatibilitě s předchozími verzemi. Další důležitou výhodou CentOSu je dobrá podpora ze strany vývojářské komunity a podobnost s komerční verzí operačního systému RHEL firmy RedHat.

Tato podobnost s operačním systémem RHEL není náhoda. V počátcích jeho vývoje dlouho předtím, než se CentOS proslavil pod svým současným jménem, bylo tento klon Red Hat Enterprise Linux označován jako cAos. V této době se jednalo o jeden z mnoha začínajících projektů s otevřeným kódem založeným na distribuci RHEL. V roce 2006 došlo k zásadnímu zlomu, když David Parsley, hlavní vývojář konkurenčního Tao Linuxu (další odnož RHELu) oznámil ukončení vývoje operačního systému Tao a přechod k systému CentOS. Tento krok znamenal pro CentOS získání veliké základny přispěvovatelů a uživatelů. V roce 2010 CentOS převzal na úkor svého největšího konkurenta Debianu vedoucí roli v oblasti webových serverů [39].

Další zásadní událostí vývoje projektu CentOS bylo oznámení společnosti Red Hat z ledna roku 2014 oficiálně sponzorovat a podporovat vývoj toho operačního systému [40]. Výsledkem bylo, že vlastnická práva byla prodána společnosti Red Hat a ten nyní zaměstnává podstatnou část hlavních vývojářů operačního systému CentOS. Přesto se projekt CentOS spoléhá na sponzorské dary uživatelů a sponzorské partnery z řad organizací užívajících tento systém pro komerční účely.

Jak již bylo uvedeno v předchozím textu, na rozdíl od RHELu je distribuce CentOS dostupná zdarma. Technická podpora je tak primárně poskytována komunitou pomocí oficiálního mail listu, webových fór a chatovacích místností. Projekt je oficiálně zastřešen společností Red Hat, nicméně klade důraz na to být otevřenější a veřejně dostupný a kompatibilní. Od verze 7.0 CentOs oficiálně podporuje pouze architektury typu x86-64.

V následující tabulce 1 jsou uvedeny pro práci relevantní technické parametry operačního systému CentOS verze 7.

Tab. 1: Vybrané technické parametry distribuce CentOS 7.

Data ukončení životnosti	
	CentOS7
Plné updaty	Q4 2020
Servisní updaty	30 Červen, 2024
Architektura	
	CentOS7
Poslední verze	7 (1511)
CPU/Paměť/Omezení filesystému(Testovaná/možná)	
	CentOS7
Maximum logických CPU	
x86_64	160/5120
Maximální paměť	
x86_64	3TB/64TB
Filesystém	
Maximální velikost souboru (ext3)	2TB
Maximální velikost file systému (ext3)	16TB
Maximální velikost souboru (ext4)	16TB
Maximální velikost file systému (ext4)	50TB

Doporučené minimální požadavky	
	CentOS7
Minimální paměť RAM	1GB/logické CPU
Minimální/doporučená velikost disku	10GB/20GB
Vlastnosti OS (Kernel, Server, Klient, další.)	
	CentOS7
Kernel báze	Linux 3.10.0
Kompilátor	GCC 4.8.5
SELinux	Ano
Ext3 Zlepšení výkonu	Ano
Databáze	MariaDB 5.5.x, PostgreSQl 9.2.x
Programovací jazyky	php 5.4, python 2.7, perl 5.16.3
Desktop GUI	Gnome 3.14, KDE 4.14
Grafika	X.org 7.7
Správce Logických Svazků (LVM)	Ano - LVM2
Kompatibilita knihoven	Ano - CentOS 5 & CentOS 6
NFS	Ano
Webový server	httpd 2.4.6 (apache)

7 REALIZACE – ZABEZPEČENÍ SÍTĚ

V této kapitole je popsána realizace síťového zabezpečení v prostředí NTT cloudu. Tato realizace má za úkol simulovat možné zabezpečení a konfiguraci cloudového úložiště z pohledu uživatele. Součástí této kapitoly je rovněž popis konfigurace virtuálního switche a zařízení INA za účelem povolení bezpečného přístupu uživatele k serverům uloženým v cloudovém prostředí.

7.1 NAT – Předklad adres

Network Address Translation (NAT) [41] je proces, při kterém firewall či router přiřazuje jinému síťovému zařízení z privátní sítě veřejnou síťovou adresu pro účely komunikace po veřejném internetu. Hlavním důsledkem užívání NATu je limitování počtu veřejných IP adres používaných organizacemi a společnostmi. Tohoto omezení je nutné zajistit z ekonomických a bezpečnostních důvodů (omezený počet veřejných adres IPv4 a úspora nákladů).

Nejrozšířenější forma síťového překladu zahrnuje velké privátní sítě používající privátní rozsahy IPv4 sítí (10.0.0.0/8, 172.16.0.0/12, 192.168.0 0/16) [42]. Adresace pomocí privátních IP funguje dobře pouze v případě interní komunikace serverů a přístrojů v rámci jedné sítě. V případě komunikace s externími zdroji (web) je však nutné, aby zařízení disponovalo vlastní veřejnou adresou, na kterou lze směrovat odpovědi na zaslané požadavky. Vzhledem k velkému množství zařízení schopných komunikovat po síti je nutné disponovat adekvátním množstvím veřejných adres. S ohledem na limitovaný počet těchto adres, je třeba aplikovat překlad adres NAT.

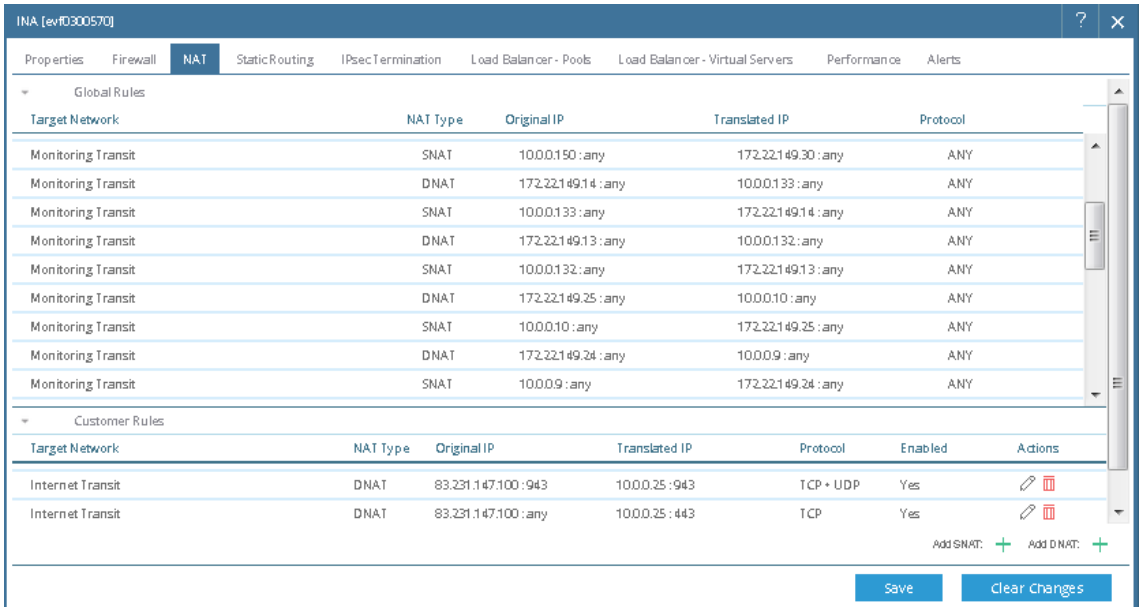
Přestože jsou operace s překladem adres NAT poměrně komplexní, probíhají tak rychle, že koncový uživatel zpravidla nezaregistruje, že samotný překlad proběhl. Klasický překlad pak probíhá následujícím způsobem. Pracovní stanice v interní síti odešle dotaz na server mino svou vlastní síť. Směrovače této sítě detekují, že požadavek je určen jiné nelokální síti a odešlou tento požadavek na externí firewall. Firewall pak provede stejný dotaz s použitím své vlastní veřejné adresy a vrátí odpověď původní dotazující stanici. Z perspektivy interního serveru se tak zdá, že komunikace proběhla přímo mezi zúčastněnými servery. Tímto způsobem lze překlad adres NAT využít pro sdílení jediné veřejné IP adresy pro stovky až tisíce uživatelů privátní sítě.

Většina moderních firewallů je typu stateful. V praxi to znamená, že uchovávají po dobu komunikace podrobné informace o spojení. Těmito informacemi bývají porty, pořadí paketů a zúčastněné IP adresy. Uchovávání těchto informací se označuje jako udržování stavu spojení. Tento stav spojení se udržuje jak pro spojení mezi firewallem a interním serverem, tak pro spojení mezi firewallem a zdrojem na internetu. Tyto informace jsou „zahozeny“ ve chvíli, kdy je ukončeno spojení.

Dalším využitím NATu je zpřístupnění služeb serverů uložených v privátní síti uživatelům z veřejné sítě. Těmto serverům je na firewallu přiřazena veřejná adresa, přes kterou pak uživatelé z veřejné sítě přistupují k serveru. Tímto způsobem byl v této práci zajištěn přístup uživatelů k serverům umístěným v cloudu pomocí openVPN klienta. Současně lze toto řešení považovat za bezpečnostní prvek. Efektivně tak totiž odděluje zařízení chráněné privátní sítě od veřejného internetu a současně představuje

bezpečnostní mezičlánek v komunikaci mezi interními servery a veřejnou sítí. V praxi se tento NAT ještě omezuje pomocí bezpečnostních pravidel omezující přístup na konkrétní porty a mapování standardních portů na nestandardní. Zažitým standardem je pak logování a ukládání záznamů příchozí komunikace na straně firewallu [43].

Obrázek 6 níže zachycuje implementovaný soubor NAT pravidel použitých v této práci. Konfigurace proběhla na zařízení INA pomocí webového portálu poskytovaného cloudovým providerem.



Obr. 6: Menu reprezentující implementované NAT pravidla.

Pravidla byla implementována za účelem umožnění přístupu uživatele k OpenVPN serveru pomocí klienta a také pro přístup k webovému portálu serveru OpenVPN pomocí protokolu HTTPS. Pro úplnost uvádím implementovaná pravidla NAT v přehledné tabulce:

Tab. 2: Tabulka překladových pravidel.

Target Network	NAT Type	Original IP	Translated IP	Protocol
Internet transit	DNAT	83.231.147.100:943	10.0.0.25:943	TCP + UDP
Internet transit	DNAT	83.231.147.100:any	10.0.0.25:443	TCP

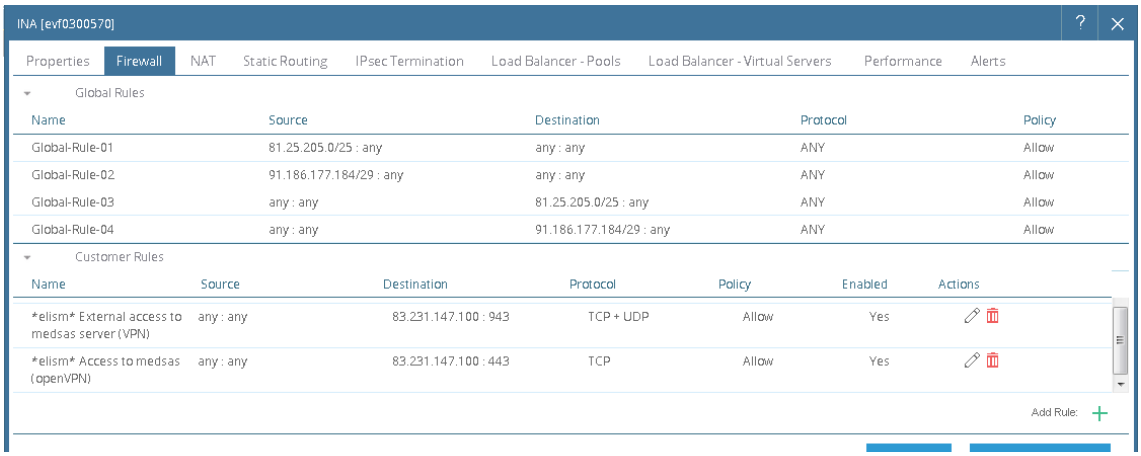
7.2 Konfigurace zabezpečení externího firewallu

Jak již bylo vysvětleno v předchozím textu, firewally slouží k ochraně privátní sítě a jejich zařízení před neoprávněným přístupem z vnější sítě. V případě této diplomové práce je externí firewall součástí cloudové infrastruktury dodávané poskytovatelem cloudového řešení. V případě NTT se jedná o zařízení INA (internet network appliance), konfigurovatelné síťové zařízení. Mezi standardní funkce tohoto síťové zařízení patří stavový firewall, překlad adres NAT, routování a konfigurace IPSec tunelů. Pro účely této diplomové práce a zajištění bezpečného přístupu k virtuálním strojům uvnitř cloudu bylo nutné implementovat následující firewallová pravidla (viz tabulka 3).

Tab. 3: Tabulka firewallových pravidel.

Rule Name	Source IP : port	Dest IP : port	Protocol	Policy
rule1	any : any	83.231.147.100:943	TCP + UDP	Allow
rule2	any : any	83.231.147.100:443	TCP	Allow

Z tabulky 3 je přehledně vidět, že přístup je povolen odkudkoli, ovšem pouze na dané porty. Pravidla byla nastavena na základě znalosti portů užívaných programem openVPN (lze změnit v konfiguraci). Obrázek 7 pak ilustruje konfiguraci firewallových pravidel pomocí portálu NTT.



Obr. 7: Příklad konfigurace firewallových pravidel v NTT.

7.3 Zabezpečení virtuálních serverů

Tato část práce se zabývá ochranou, jež byla implementována přímo na jednotlivých serverech. Kromě ochrany samotných serverů proti potencionální útokům zvenčí tato konfigurace zároveň zajišťuje odstínění a izolaci řešení od zbytku cloudového prostředí v dané virtuální místnosti.

Ochrana byla realizována pomocí standartních iptables [44], které jsou součástí každé distribuce operačního systému CentOS. Pravidla byla konfigurována postupně, podle požadavků instalovaných softwarových součástí cloudového úložiště v souladu s původním návrhem. Cílem bylo zajistit bezpečnou nepřerušovanou konektivitu mezi přístupovým a autentizačním serverem a současně znemožnit komunikaci s ostatními virtuálními stroji v daném cloudovém úložišti.

Následuje ukázka použitých firewallových pravidel iptables na straně přístupového serveru:

```
    service iptables status
    Table: filter
    Chain INPUT (policy DROP)
    num  target     prot opt source               destination
    1    ACCEPT     all  --  0.0.0.0/0           0.0.0.0/0           state
RELATED,ESTABLISHED
    2    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           tcp
dpt:53
    3    ACCEPT     udp  --  0.0.0.0/0           0.0.0.0/0           udp
dpt:53
    4    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           tcp
dpt:22
    5    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           tcp
dpt:443
    6    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           tcp
dpt:943
    7    ACCEPT     47   --  0.0.0.0/0           0.0.0.0/0
    8    ACCEPT     esp  --  0.0.0.0/0           0.0.0.0/0
    9    ACCEPT     icmp --  0.0.0.0/0           0.0.0.0/0
    10   INNER      all  --  0.0.0.0/0           0.0.0.0/0
    11   INNER     all  --  0.0.0.0/0           0.0.0.0/0           policy
match dir in pol ipsec mode tunnel
    12   GLOBAL     all  --  0.0.0.0/0           0.0.0.0/0

    Chain FORWARD (policy ACCEPT)
    num  target     prot opt source               destination

    Chain OUTPUT (policy ACCEPT)
    num  target     prot opt source               destination
    1    ACCEPT     all  --  10.0.0.140          0.0.0.0/0           state
RELATED,ESTABLISHED
    2    ACCEPT     all  --  10.0.0.140           0.0.0.0/0
    3    DROP          all  --  0.0.0.0/0              0.0.0.0/0

    Chain GLOBAL (2 references)
    num  target     prot opt source               destination
    1    LOG        all  --  0.0.0.0/0           0.0.0.0/0           LOG
flags 0 level 4 prefix `iptables: '
    2    DROP       all  --  0.0.0.0/0           0.0.0.0/0
```

```
    Chain INNER (2 references)
    num  target     prot opt source               destination
    1    ACCEPT     all  --  0.0.0.0/0           0.0.0.0/0           state
RELATED,ESTABLISHED
    2    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
    3    ACCEPT     tcp  --  10.0.0.254          0.0.0.0/0            tcp
dpt:22 state NEW
    4    ACCEPT     tcp  --  10.0.0.140          0.0.0.0/0            tcp
dpt:1812,1813 state NEW
    5    ACCEPT     tcp  --  10.0.0.140          0.0.0.0/0            tcp
dpt:22 state NEW
    7   GLOBAL     all  --  0.0.0.0/0            0.0.0.0/0
```

8 ARCHITEKTURA A PRINCIPY

Tato kapitola popisuje zvolenou architekturu, principy, použitý software a jeho konfiguraci v realizaci návrhu cloudového úložiště. Rovněž jsou zde vysvětleny důvody, které vedly k výběru tohoto způsobu řešení. Popsán je princip autentizace při navázání spojení mezi uživatelem a přístupovým serverem a samotná realizace přístupu uživatele. V neposlední řadě jsou také objasněny rozdíly v realizaci oproti původnímu návrhu a uvedeny možná budoucí vylepšení zabezpečení cloudového úložiště.

8.1 Přístup do cloudu

Zásadním krokem, kterým se provozovatel cloudového úložiště musí zabývat je bezpečný šifrovaný přístup uživatelů k serverům v tomto cloudu. Poskytovatel musí zajistit důvěrnost, autenticitu a integritu spojení. Standardně se tedy využívá bezpečných šifrovaných VPN spojení poskytovaných různými komerčními programy. V případě této realizace jsem se rozhodl využít veřejně dostupného programu OpenVPN [45]. Podobně jako u konkurenčních programů pracuje OpenVPN na bázi server-klient. Jak již bylo uvedeno v návrhu řešení, OpenVPN server byl tedy umístěn na přístupový, neboli centrální server, který je označen jako medscs.

Výhody této varianty spočívají zejména v její bezplatné distribuci, aktuálnosti a poměrně kvalitní a dobře dostupné technické podpoře a dokumentaci. Další výhodou této možnosti pak je podpora autentizace vůči serveru RADIUS [46], jež byla součástí původního návrhu cloudového úložiště. V následujícím textu tedy bude uvedena stručná charakteristika programu OpenVPN a popsáno řešení přístupu uživatele do cloudového úložiště pomocí OpenVPN klienta.

8.1.1 OpenVPN

Jak již bylo zmíněno, program OpenVPN je veřejně dostupný software sloužící pro vytváření bezpečných VPN spojení pro komunikaci mezi uživatelem a druhou stranou (point-to-point nebo site-to-site). Tento program je od roku 2001 vyvíjen společností OpenVPN Technologies.

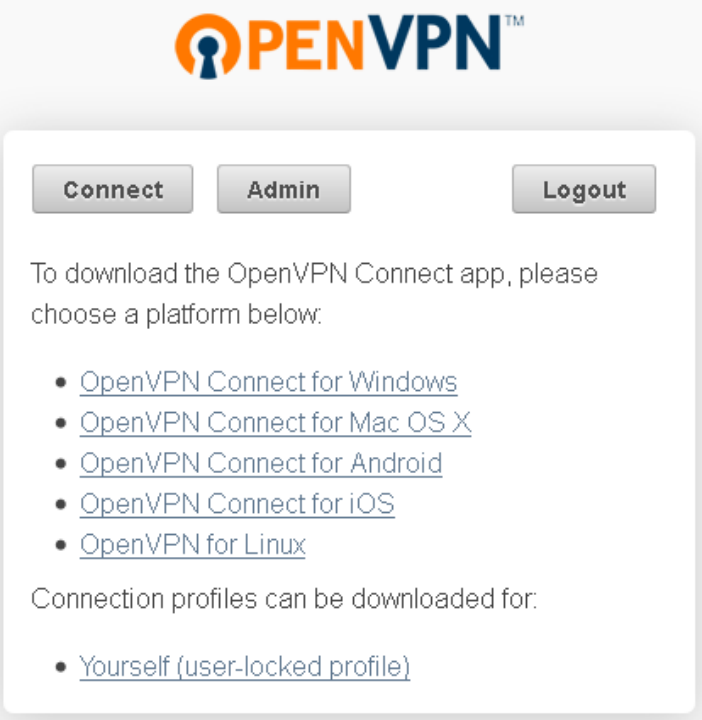
Samotný program využívá vlastní implementaci bezpečnostních protokolů SSL/TLS pro výměnu klíčů. Program je schopen vytvářet VPN spojení a komunikovat s druhou stranou i v případě užití problematického překladu adres NAT. OpenVPN umožňuje autentizaci svých peerů pomocí předsdíleného tajného klíče, certifikátů, či kombinace uživatelského jména a hesla. Současně je OpenVPN možné použít v kombinaci s certifikační autoritou, která přiděluje klientům autentizační certifikáty. Implementace programu samotného vychází z kryptografické knihovny OpenSSL a ze standardu protokolu SSLv3/TLSv1 [45].

Funkce této knihovny poskytují veškeré šifrovací možnosti dostupné pro OpenSSL balík. Šifrovány tak jsou data i řídící kanály. Rovněž je možno nakonfigurovat mechanizmus HMAC paketové autentizace jako dalšího prvku bezpečnosti. OpenVPN umožňuje podporu hardwarové akcelerace pro získání optimálnějšího výkonu pro účely šifrování [47].

Jak již bylo uvedeno výše, OpenVPN poskytuje široké možnosti autentizace. Během realizace návrhu byla použita metoda ověření pomocí dvojice uživatelského jména a hesla klientem vůči serveru (viz obrázek 8). Tento přístup byl změněn po instalaci autentizačního serveru medsas. Na tomto serveru se nachází instance RADIUS serveru, která slouží jako autentizační autorita klientům usilujícím o přístup do cloudového úložiště.

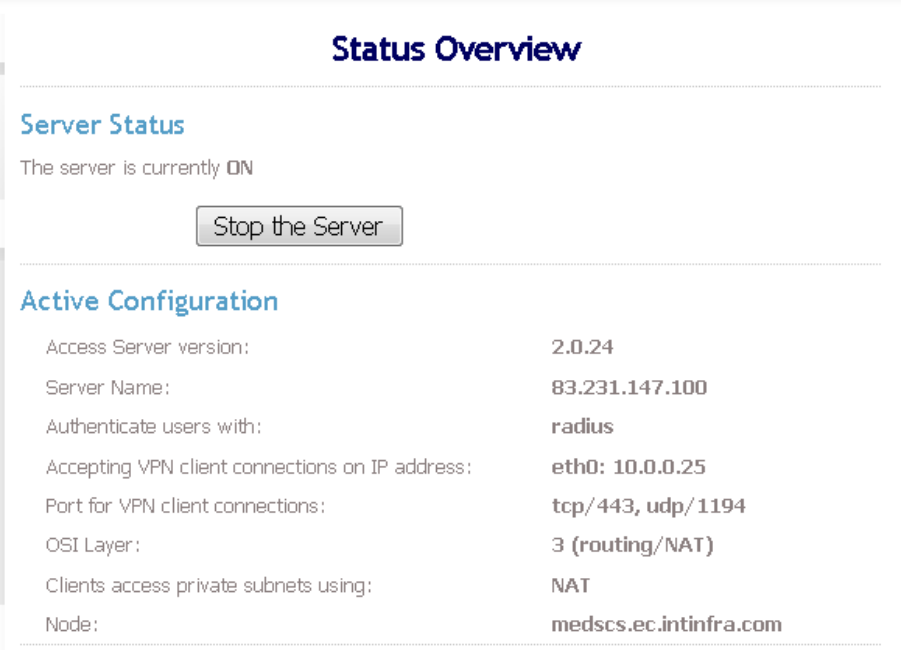
Obr. 8: Ukázka autentizačního dialogu OpenVPN klienta.

Samotný OpenVPN klient může být distribuován serverem pomocí webové stránky (HTTPS na port 943). Uživateli pak je nabídnut klient v závislosti na použitém operačním systému. Zde má uživatel stáhnout klienta, pomocí kterého se přihlašuje k serveru, respektive cloudovému úložišti.



Obr. 9: Příklad distribuce OpenVPN klienta.

Program OpenVPN může používat jak protokol UDP, tak TCP, přičemž využívá multiplexování SSL tunelů na jediný TCP/UDP port [45]. Od verze 2.3 OpenVPN plně podporuje IPv6, to znamená přenos uvnitř VPN tunelu a současně navázání VPN spojení na IPv6 síti [46]. Nesporná výhoda užití standardních protokolů (TCP a UDP) je možnost nasazení těchto VPN tunelů v případech kdy ISP blokuje standardní IPSec protokoly.



Obrázek 10: Příklad základní konfigurace OpenVPN serveru.

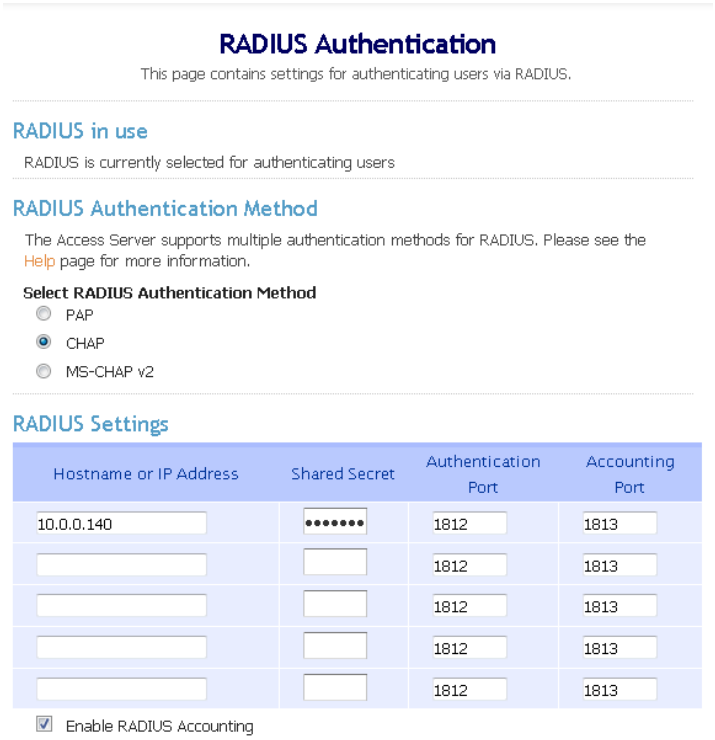
OpenVPN může být rozšířen pomocí pluginů, či skriptů, které jsou pak volány na základě prováděných akcí. Příkladem může být užití autentizačního serveru RADIUS (viz obrázek 11) pro potřeby pokročilejší autentizace a logování přístupu. Dalším možným vylepšením může být rozšíření serveru o databázi MySQL, či LDAP, jež poskytují infrastrukturu pro efektivní logování pokusů o přístup do interní sítě. Seznam veškerých dostupných pluginů a modulů lze nalézt na odpovídajících wiki stránkách OpenVPN komunity [48].

8.1.2 FreeRADIUS

Jak již bylo zmíněno v předcházejících kapitolách, součástí realizace byl autentizační server medsas, poskytující autentizační autoritu prostřednictvím serveru RADIUS. Pro tento účel byla použita veřejně dostupná implementace FreeRADIUS.

FreeRADIUS je v současné době nejpopulárnější veřejně dostupný RADIUS server a jako takový je proto nejčastěji používaný RADIUS server na světě. Podporuje veškeré standardní autentizační protokoly. Kromě samotné funkcionality FreeRADIUS nabízí uživateli webové rozhraní dialupadmin sloužící k pohodlné správě serveru. Jakožto spolehlivý poskytovatel AAA (autentizace, autorizace, správa účtů), stal se FreeRADIUS součástí mnoha vestavěných systém zaručující správu přístupu k síti. Velmi často je tak tento software využíván telekomunikačními společnostmi, poskytovateli internetového připojení a dalšími [49].

Mezi standardně podporované moduly pak lze řadit LDAP, MySQL, PostgreSQL, Oracle a mnohé další databáze [49]. FreeRADIUS rovněž podporuje široké spektrum EAP autentizačních typů, například PEAP a EAP-TTLS. Od verze 2.0.0 podpora zahrnuje sítě IPv6 a VPMS (VLAN Management Policy Server).



Obr. 11: Ukázka konfigurace CHAP autentizace OpenVPN serveru vůči RADIUS serveru.

8.2 Autentizace

V této podkapitole bude vysvětlen a popsán princip autentizace použitý v práci. Dále je zde uveden příklad výstupu při autentizaci uživatele a ověřování kombinace jeho uživatelského jména a hesla serverem RADIUS.

Jak jíž bylo uvedeno v textu výše, uživatel přihlašující se do systému pomocí klienta OpenVPN musí provést úspěšnou autentizaci vůči RADIUS serveru. V případě této práce jsem se rozhodl uchovávat přihlašovací údaje bezpečně pomocí databáze. Mezi použitelné varianty implementace patřily databáze PostgreSQL, Oracle a MySQL.

Jako první byla vyloučena velmi robustní databáze Oracle. Důvodem byla především nutnost zakoupení komerční licence, předchozí neznalost práce s touto databází. Dalším důvodem vyloučení této možnosti byla vysoká náročnost na výpočetní zdroje.

Další prověřovanou variantou bylo použití v CentOSu vestavěné databáze PostgreSQL. Tato možnost byla po několika neúspěšných pokusech zavrhnuta ve prospěch databáze MySQL.

Pro databázi MySQL jsem se rozhodl vzhledem k předešlé zkušenosti, relativní jednoduchosti instalace a dobrým výkonnostním výsledkům. Oproti PostgreSQL se ukázala instalace, konfigurace a následující integrace se serverem RADIUS jako poměrně jednoduchá a bezproblémová. Následující ukázka zachycuje proces autentizace uživatele uživatelským jménem a heslem vůči serveru RADIUS.

```
   root@medsas:~#  radtest  -x  test  test  127.0.0.1  0
test123XX!
   Sending Access-Request of id 248 to 127.0.0.1 port 1812
           User-Name = "test"
           User-Password = "test"
           NAS-IP-Address = 10.0.0.140
           NAS-Port = 0
           Message-Authenticator                          =
0x00000000000000000000000000000000
   rad_recv: Access-Accept packet from host 127.0.0.1 port
1812, id=248, length=20
```

Jak je patrné z ukázky, jako první je zaslána serveru žádost o přístup klientem. Součástí této žádosti je klientovo uživatelské jméno a heslo. Po ověření zaslaných údajů a porovnání s daty uloženými v databázi MySQL server odpovídá klientovi a povoluje mu přístup. Jako metoda autentizace byla po konzultaci s vedoucím diplomové práce zvolena metoda CHAP. Tento typ autentizace bude popsán podrobněji v podkapitole 8.2.2.

8.2.1 MySQL

MySQL je veřejně přístupná relační databáze pracující na principu klient-server. Podobně jako tomu je v případě operačního systému CentOS a jeho vývoje firmou RedHat, tak i MySQL byl původně produktem soukromé komerční společnosti a po svém zveřejnění a rozšíření byl odkoupen konkurenční společností Oracle. Ta nyní poskytuje MySQL pro soukromé účely zdarma, zatímco pro užití v komerčním sektoru nabízí různé

kustomizované placené verze. Společnost Oracle tak nabízí uživatelům placených verzí kvalitní technickou podporu.

V současné době je MySQL hojně využíván jako databáze pro webové aplikace. Velmi často se tak používá ve spojení s webovým nástrojem Linuxových distribucí, programem Apache a programovacími jazyky PHP, PERL a Python. [50] Možnou nevýhodu oproti jiným databázím lze spatřovat v neexistenci grafického rozhraní pro správu databáze. Tuto potencionální nevýhodu však řeší nástroje poskytované dalšími výrobci.

Samotná databáze MySQL je implementována pomocí jazyka C a C++. Technická podpora je dostupná prostřednictvím oficiálního manuálu, chatů a fór. Placená podpora nabízená společností Oracle se vztahuje pouze na Enterprise verze. Alternativní možností pak mohou být databáze vycházející z původní MySQL – Percona, MariaDB, TokuDB a další.

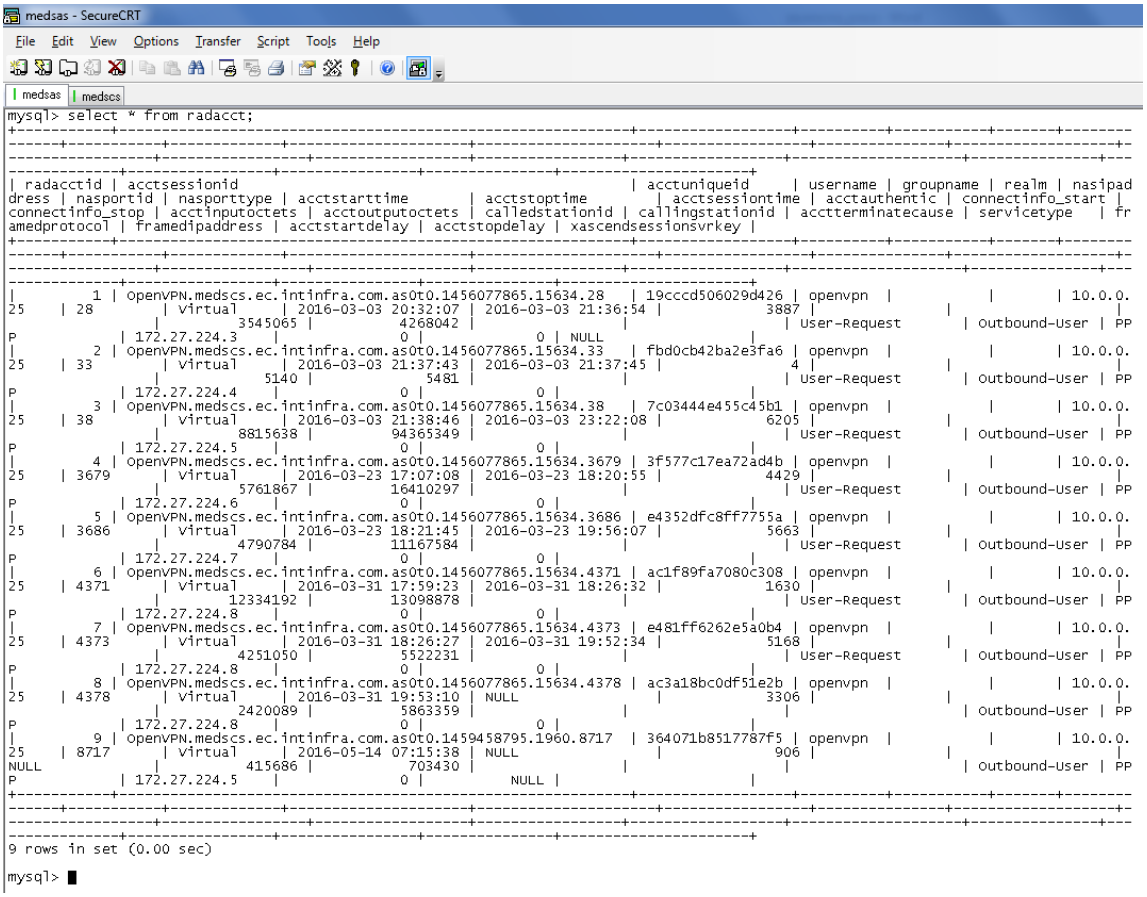
Následující krátký příklad ilustruje práci s MySQL databází za účelem zobrazení záznamů o přístupu pomocí RADIUS serveru. Zobrazení databází:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| radius             |
+--------------------+
3 rows in set (0.00 sec)
```

Volba databáze a zobrazení uložených tabulek:

```
mysql> use radius;
Database changed
mysql> show tables;
+------------------+
| Tables_in_radius |
+------------------+
| radacct          |
| radcheck         |
| radgroupcheck    |
| radgroupreply    |
| radpostauth      |
| radreply         |
| radusergroup     |
+------------------+
7 rows in set (0.00 sec)
```

Pomocí dotazu SELECT * FROM redacct je v dalším kroku zobrazen seznam přístupů uživatelů zalogovaných serverem RADIUS (viz obrázek 12).



Obr. :12 Výstup z databáze. Přehled uložených přístupů autentizovaných serverem RADIUS.

8.2.2 CHAP

CHAP - Challenge Handshake Authentication Protocol [51] je protokol, který autentizuje uživatele vůči autentizační entitě. Touto entitou může být například poskytovatel internetového připojení. V našem případě bude jako autentizační entita sloužit server RADIUS. Samotný CHAP vyžaduje, aby klient i server disponovali znalostí hesla. Výhodou oproti alternativnímu řešení pomocí protokolu PAP [52] je, že nedochází k přenosu hesla po síti.

Protokol CHAP pak pracuje na následujícím principu. Jakmile je ustaveno spojení, dojde k zaslání žádosti o autentizaci jedním z komunikujících peerů. V případě PAP by pak došlo k přenosu uživatelského jména a hesla (ideálně šifrovanému) a porovnání na straně serveru. Tento přístup má zjevnou slabinu v přenosu hesla po síti. V případě CHAP autentizační server zašle náhodně generovanou výzvu (vygenerovaný řetězec) klientovi. Klient použije své uživatelské jméno pro vyhledání hesla a to zkombinuje daným způsobem s přijatým řetězcem. Tuto kombinaci pak zahešuje a vrátí serveru společně s uživatelským jménem. Na straně serveru pak dojde k vytvoření odpovídajícího hashe pomocí odeslaného náhodného řetězce a pomocí odpovídajícího hesla uživatele, jenž má uložené v databázi a k následnému porovnání takto vytvořeného řetězce (hashe)

a obdrženého řetězce. Toto ověřování je pak opakováno v pravidelných časových intervalech. Následující příklad ilustruje proces neúspěšné autentizace pomocí protokolu CHAP, která je následně zamítnuta serverem RADIUS.

```
   root@medsas:/etc/raddb# radtest -t chap -x bob test3
127.0.0.1 0 test123!
   Sending Access-Request of id 1 to 127.0.0.1 port 1812
           User-Name = "bob"
           CHAP-Password                                  =
0x01ca9cded2b1be7c9d446b54787dcdcb29
           NAS-IP-Address = 10.0.0.140
           NAS-Port = 0
           Message-Authenticator                          =
0x00000000000000000000000000000000
   rad_recv: Access-Reject packet from host 127.0.0.1 port
1812, id=1, length=20
```

Následující příklad naopak ilustruje úspěšnou autentizaci pomocí protokolu CHAP, která je následně povolena serverem RADIUS.

```
   root@medsas:/etc/raddb# radtest -t chap -x bob test
127.0.0.1 0 test123!
   Sending Access-Request of id 26 to 127.0.0.1 port 1812
           User-Name = "bob"
           CHAP-Password                                  =
0x1a83e5d7ec2e485a15cb49be5164ea12d6
           NAS-IP-Address = 10.0.0.140
           NAS-Port = 0
           Message-Authenticator                          =
0x00000000000000000000000000000000
   rad_recv: Access-Accept packet from host 127.0.0.1 port
1812, id=26, length=20
```

8.3 Rozdíly oproti návrhu

Hlavním rozdílem oproti původnímu návrhu bylo nahrazení GRE tunelů IPSec tunely postavenými pomocí programu OpenSwan [53]. Důvodem k této změně byl požadavek diplomové práce otestovat dostupné šifrovací algoritmy a ověřit jejich chování v závislosti na zvolené délce šifrovacího klíče. Toto nastavení jako takové není možné ovlivnit v konfiguraci GRE tunelů. Jako alternativní metodu umožňující simulovat přenos souborů pomocí různých kryptografických algoritmů bylo zvoleno použít IPSec tunely realizované pomocí programu OpenSwan.

Tento program umožňuje manuálně konfigurovat algoritmy použité v první a druhé fázi výstavby IPSec tunelu, přičemž součástí nastavení je i konfigurace užitého klíče. Realizace pomocí programu OpenSwan je podrobněji popsána v kapitole deset, stejně jako průběh testování a analýza dosažených výsledků.

9 EFEKTIVNÍ UKLÁDÁNÍ DAT

9.1 OpenDedup

Jak je již z názvu patrné, jedná se o veřejně dostupný software. Účel tohoto programu je šetřit místo na fyzickém disku serverů a virtuálních strojů pomocí takzvané deduplikace. Deduplikací souborového systému rozumíme proces, kdy jsou vyhledávána redundatní data, tedy data která již na daném serveru mají jinou kopii. Cílem tohoto procesu je dosáhnout stavu, kdy na serveru uchováváme pouze jedinou kopii daného souboru, přičemž všechny ostatní kopie tohoto souboru jsou reprezentovány odkazem na tuto jedinou kopii [54]. Jako takový tedy proces deduplikace pracuje jako virtuální souborový systém spravující ukládaná data.

Tímto způsobem je redukováno spotřebované místo na disku. Jak je patrné, míra efektivity této úspory je závislá na úspěšnosti systému detekovat duplikovaná data. Cenou za tuto úsporu je však zvýšená utilizace CPU, jež neustále vyhledává potencionální duplikovaná data. Tato spotřeba výpočetního výkonu může vést ke snížené rychlosti zápisu a čtení.

V praxi existuje velké množství deduplikačních programů. Zásadním rozdílem je však doba, kdy dochází k samotné deduplikaci. První možností je typ post-process, kdy k provedení procesu deduplikace dochází po uložení celého souboru na disk. Druhou možností je in-line deduplikace. V tomto případě dochází k vyhledání duplikátních dat před samotným uložením souboru na disk serveru. Data, která tak byla vyhodnocena jako duplikátní jsou uložena na disk pouze jednou. OpenDedup umožňuje obojí možné varianty. V našem případě jsem se rozhodl použít variantu post-process. Důvodem byla snaha zamezit zkreslení změřených výsledků dob přenosu procesem kontroly duplikátních dat.

Dalším důležitým rozdílem mezi deduplikačními systémy je způsob, jakým je rozpoznáván duplikátní soubor. Nejjednodušší metodou bývá porovnání celých souborů, nazývané single-instance úložiště. Další možností je rozdělení souborů do kratších úseků, takzvaných chunků a jejich následné porovnávání. V tomto projektu byla z důvodu minimálnosti řešení a s ohledem na celkový výkon systému použita první z popsaných možností.

Samotný OpenDedup je implementován pomocí programovacího jazyka Java a softwaru FUSE (Filesystem in Userspace), jež zajišťuje komunikaci s kernelem a zprostředkovává tak programu funkce jádra operačního systému [54].

9.2 Thin provisioning

Thin provisioning je metoda alokování výpočetních zdrojů ve virtuálních technologiích vedoucí k optimalizaci spotřeby těchto zdrojů. Tato technologie umožňuje virtuálním serverům alokovat více virtuálních zdrojů, než je aktuální kapacita fyzického hardware. Princip tohoto přidělování spočívá v krátkodobém přidělování výpočetní kapacity v závislosti na aktuální potřebě. Opakem tohoto přístupu je takzvaný thick provisioning, tedy dopředného alokování výpočetních zdrojů.

„Tenký“ provisioning bývá nejčastěji aplikován na diskovou kapacitu, ale stejně tak může být použit v případě operační paměti RAM, či CPU. Příkladem může být alokování paměti procesům prováděnými počítači. Jednotlivé procesy jednají, jako by měly alokovány reálnou paměť dané velikosti, přičemž součet veškeré alokované paměti je vyšší než skutečná dostupná paměť [55].

Efektivita zvoleného přístupu se pak odvíjí podle způsobu využití serveru. Metoda „tlustého“ provisioning je obvykle efektivní v případech, kdy se množství použitých výpočetních zdrojů blíží množství zdrojů rezervovaných. Naopak v případě, že je využívána pouze malá část alokovaných zdrojů, thin provisioning se ukazuje jako efektivnější. Díky tomuto přístupu je tak možné dosáhnout výrazné úspory pomocí vysokého využitím dostupných alokovaných výpočetních zdrojů.

V případě této práce a poskytovatele cloudového řešení NTT, uživateli není umožněno manuálně změnit typ provisioningu. Z dokumentace jsem se dozvěděl, že standardně je systém provisionován jako tenký. V případě potřeby vytvoření virtuálního stroje s jinou konfigurací, je třeba vyhledat technickou podporu NTT.

10 SROVNÁNÍ AUTENTIZAČNÍCH METOD

V této kapitole se práce zabývá bezpečným propojením přístupového a autentizačního serveru pomocí IPSec tunelu realizovaném programem OpenSwan za účelem simulace bezpečného přenosu dat mezi servery v závislosti na užitém šifrovacím algoritmu a zvolené délce klíče. Dále je pak v textu popsán samotný program, příklad použité konfigurace. V následující sekci jsou popsána testovací data a postup užitý pro získání statistických dat pro potřeby srovnání výsledků přenosů dat pomocí různých šifrovacích algoritmů. V poslední části jsou přehledně prezentovány dosažené výsledky.

10.1 OpenSwan

OpenSwan je název veřejně dostupného, otevřeného softwarového VPN řešení, umožňujícího uživatelům budovat rozsáhlé bezpečné VPN sítě napříč širokým spektrem technologií a použitého hardware. Historie tohoto VPN řešení sahá do doby, kdy komerční společnosti využívali předchůdce dnešního OpenSwanu, programu Freeswan jako jednoho z klíčových prvků testování a prezentace interkonektivity s různými technologiemi a hardwarovým vybavením. V této době bylo zásadním krokem přidání podpory protokolu X.509. Následoval neméně důležitý patch, který umožnil podporu NATování a překlenutí s ním spojených problémů, takzvaný NAT traversal. Tato dvě rozšíření znamenala obrovský úspěch a expanzi do komerčního sektoru a následnou standardizaci použitých technologií a principů.

Kvůli řadě právních a technologických rozporů mezi původním vlastníkem a vývojáři rozšiřujícími otevřený kód, došlo v roce 2003 k vzájemné dohodě a rozštěpení projektu na původní Freeswan a nový komerčně zaměřený OpenSwan. Oproti svému předchůdci, tak nový OpenSwan postrádal některé z původních sekcí kódu, avšak obsahoval zmíněné patche a rozšíření pro stavbu VPN sítí.

Toto rozdělení tak umožnilo zaměřit se na výzkum a implementaci nových VPN technologií. Jako záštita projektu OpenSwan byla založena firma Xelerance, jejíž primárním cílem bylo poskytovat placenou podporu zákazníkům z které byl následovně financován vývoj projektu. Bez zátěže a omezení původního majitele a záměru Freeswanu, OpenSwan postupně převzal vedoucí roli Freeswanu a rozšířil se do nejznámějších Linuxových distribucí jakými jsou RedHat Enterprise Linux, Debian, SuSe [53].

V průběhu dalších let byl OpenSwan nadále rozšiřován dle aktuálních požadavků komerčních zákazníků. Paradoxně, přes vysokou úspěšnost, oblíbenost v komerčním sektoru a částečně kvůli otevřenosti kódu se firma dostala koncem roku 2011 do problémů. IPsec a IKE implementované v projektu byly totiž tak spolehlivé, že jen malý zlomek uživatelů požadoval funkcionalitu novějších verzí. Finanční situaci projektu OpenSwan také nezlepšila dostupnost placené technické podpory dodávané firmou RedHat. V současné době tak je možné transformovaný projekt OpenSwan hledat pod názvem Libreswan [53].

Hlavní konfiguraci VPN řešení OpenSwan lze nalézt v souboru ipsec.conf. V tomto souboru je možné nastavit typ IPSec spojení (transportní nebo tunelový), šifrovací algoritmus pro každou fázi, klíč, autentizace, platnost použitých klíčů a rovněž jsou zde

definovány koncové body (peers) VPN tunelů. Samotný konfigurační soubor je rozdělen na sekce, přičemž klíčové parametry jsou uloženy v sekci conn [56]. Následující příklad ilustruje konfigurace OpenSwanu na přístupovém serveru medscs.

```
   conn test
            auto=start                                    #any
reboot causes immediate renegotiation
            type=transport                          #transport
mode ipsec
            authby=secret
#authentication
            ike=3des-sha1-modp1024       #phase 1 aka isakmp
sa
            ikelifetime=8h                             #phase
1 sa lifetime
            esp=3des-sha1                            #phase 2
aka ipsec sa
            keylife=1h                                 #phase
2 sa lifetime
            pfs=no
            ###our gateway
            left=10.0.0.25                               #the
IP address of the local IPSec peer
            leftnexthop=10.0.0.25                    #default
gateway
            leftprotoport=47                        #match the
GRE traffic, this line is very important
            ###remote peer
            right=10.0.0.140                             #the
IP address of the remote IPSec peer
            rightnexthop=10.0.0.140            #peer default
gateway
            rightprotoport=47                       #match the
GRE traffic
```

10.2 Testovací data a princip testování

Tato podkapitola se zabývá samotnými testovacími daty a způsobem provedení testů. Prvotním úkolem pro získání statistických dat bylo vytvoření prostředí vhodné pro testování šifrovacích algoritmů. V případě cloudového řešení jsem se pak rozhodl použít dva virtuální stroje propojené VPN tunelem. Podmínkou tohoto VPN tunelu byla možnost měnit použitý šifrovací algoritmus a délku klíče. Tuto část zajistilo použití programu OpenSwan.

Dalším úkolem bylo vytvoření testovacích souborů, které budou kopírovány tunelem VPN a ukládány na druhém serveru. K tomuto účelu jsem vygeneroval pomocí příkazu dd soubor output.dat o velikosti 100 MB.

```
   root@medscs:/tmp# dd if=/dev/zero of=output.dat  bs=100M
count=1
   root@medscs:/tmp# ls -lhtr
   total 101M  -rw-r--r--  1  root  root  100M  May  14  13:57
output.dat
```

Pro získání podrobnějších záznamů o proběhlém přenosu jsem na zúčastněné servery nainstaloval jednoduchý monitorovací nástroj vnstat. Tento nástroj sleduje síťový provoz na definovaných rozhraních a zapisuje získaná data do vlastní databáze. Následuje příklad výstupu nástroje vnstat pro definovaný tunel tun_test.

```
   root@medscs:/tmp# vnstat
    tun_test:
          May '16        659 KiB   /   100.70 MiB   /   101.35
MiB   /       0 KiB
            today        659 KiB   /   100.70 MiB   /   101.35
MiB   /       --
```

Samotné testování algoritmů pak probíhalo následovně. Pro každou kombinaci dostupného šifrovací algoritmu (3DES, AES, Blowfish, Twofish, Serpent, Cast), hashovací funkce (MD5, SHA-1, SHA-512) a délky klíče (128 nebo 256 bitů) jsem provedl deset kopírování souboru mezi servery. K tomuto kopírování jsem vytvořil jednoduchý skript. Tento skript pak byl spuštěn postupně pětkrát, pro zajištění dostatečného množství relevantních dat a snížení odchylky způsobené chováním sítě. Veškeré vytvořené pomocné skripty je možné nalézt v příloze 1. Z uložených dat zachycených programem vnstat jsem pak následovně vyexportoval celkové délky jednotlivých přenosů dat. Z těchto hodnot jsem pak vypočítal průměrnou délku a dobu přenosu. Na základě těchto parametrů pak proběhlo konečné srovnání testovaných šifrovacích algoritmů.

Pro veškeré testy jsem IPSec fázi jedna (ustavení tunelu - IKE) ponechal nastavenou na ike=3des-sha1;modp8192. Tento zápis programu OpenSwan přikazuje použít šifrovací algoritmus 3DES, přičemž použitá hashovací funkce je typu SHA1. Poslední parametr udává takzvanou Diffie-Hellman skupinu. Kryptografický algoritmus Diffie Hellman slouží k bezpečné výměně šifrovacích klíčů po nezabezpečeném kanále. Typ skupiny pak určuje počet bitů, jenž je použit jako exponent pro výpočet bezpečného klíče. Tento parametr jsem pro účely testování ponechal ve všech případech nastaven na hodnotu modp8192, protože se během experimentování ukázal vliv tohoto parametru jako zanedbatelný.

Podobným způsobem pak probíhá nastavení parametrů šifrování přenosu, tedy fáze dvě ustavení IPSec tunelu. Jako ověřovací klíč byl použit elektronický podpis RSA vygenerovaný na komunikujících systémech. Pro účely hashování byly použity hashovací algoritmy MD5, SHA1 a SHA2-512.

10.3 Výsledky srovnání

Prvním algoritmem, který byl podroben testování byl algoritmus 3DES. Samotný algoritmus 3DES je založen na kaskádovém užití implementace algoritmu DES. DES je

již dnes prolomený standard pro šifrování dat vyvinutý v sedmdesátých letech. Délka klíče tohoto algoritmu je v dnešní době nedostačující a samotný algoritmus již byl prolomen útokem typu brute-force. Samotný 3DES je v současné době považován za dostatečně bezpečný. Díky své složitosti je však 3DES v porovnání s ostatními algoritmy velmi pomalý. Získané výsledky ukázaly, že šifrovací algoritmus byl nejpomalejším z testovaných algoritmů, dosahující rychlosti kolem 11MB/s. Program OpenSwan v případe algoritmu 3DES neumožňuje změnit délku klíče, tudíž testování proběhlo s klíčem o délce 192 bitů.

Dalším testovaným algoritmem byla šifra CAST. Tato šifra byla vyvinuta jako potencionální nástupce algoritmu DES. CAST používá 64 bitové bloky a klíče o délce 40 nebo 128 znaků. Samotný algoritmus je pak založen na principu Feistelovy šifry. Ve srovnání s ostatními porovnávanými algoritmy tento dosahoval rychlosti přenosu v rozmezí 24-27 MB/s. Rovněž tato šifra má pevně nastavenou délku klíče (128 bitů).

Jako třetí jsem se pokusil otestovat algoritmus Blowfish vyvinutý v roce 1993. Tento algoritmus zůstává stále neprolomen a nabízí poměrně dobrou rychlost šifrování. Algoritmus byl vyvinut jako možná alternativa stárnoucí DES šifry. Z hlediska architektury se pak podobá šifře CAST. Rozdílem pak je délka klíče, která se v případě Blowfish může pohybovat v rozmezí od 32 až 448 bitů. Tento algoritmus se bohužel nepodařilo otestovat, navzdory informacím uvedeným v dokumentaci projektu OpenSwan. Jako potencionální zdroj problému se jeví použití nevhodného kernelu (jádra) operačního systému během kompilace programu OpenSwan. Algoritmus Blowfish byl následně nahrazen algoritmem Twofish.

Šifra Twofish je nástupcem algoritmu Blowfish. Jako takový sdílí se svým předchůdcem strukturu podobnou Feistelově šifře. Délka klíčů v případě algoritmu Twofish bývá 128 nebo 256 bitů. Získané výsledky pak ukázaly, že v případě použití klíče délky 128 bitů byl algoritmus druhý nejrychlejší. Dosahovaná přenosová rychlost se pohybovala kolem 30 MB/s.

Pátým testovaným algoritmem byl algoritmus Serpent. Tento algoritmus byl opět vytvořen jako potencionální náhrada zastaralého algoritmu DES. Délka klíče algoritmu Serpent může být nastavena na 128, 192 a 256 bitů. Oproti ostatním srovnávaným algoritmům byla šifra Serpent implementována jako paralelní algoritmus. Nechtěným důsledkem tohoto paralelismu je náchylnost vůči různým formám kryptoanalýzy a z toho vyplývající potencionální riziko prolomení šifry. Z hlediska naměřených výsledků algoritmus Serpent dosahoval rychlosti kolem 22 MB/s při délce klíče 128 bitů.

Jako nejrychlejší ze srovnávaných algoritmů se ukázal být algoritmus AES. S výjimkou případu, kdy bylo užito hashovací funkce SHA1 bylo šifrou AES pravidelně dosahováno nejvyšších rychlostí přenosů dat. Jak již bylo nastíněno AES je nástupcem algoritmu DES. Jako takový je standardem symetrického šifrování dat. Délka klíčů může být 128, 192, či 256 bitů. Algoritmus používá bloky o 128 bitech. Kompletní tabulku dosažených výsledků lze nalézt v příloze 2 na konci dokumentu.

V případě algoritmů AES, Serpent a Twofish bylo možné provést měření dob přenosu v závislosti na odlišné délce použitých klíčů. Podle očekávání výsledky ukázaly klesající tendenci přenosové rychlosti odvíjející se od zvyšující se délky použitého šifrovacího klíče.

Zajímavé se ukázalo rovněž srovnání algoritmů z hlediska kryptografické síly.

Z testovaných kombinací šifrovacího algoritmu, délky klíče a hashovací funkce pak lze prohlásit za nejslabší kombinaci šifry 3DES a hashovací funkce MD5 za použití klíče o délce 128 bitů [57]. Další potencionálně slabou šifrou se ukázal být algoritmus CAST. Tento algoritmus během procesu šifrování využívá 64-bitové bloky dat, což z hlediska kryptoanalýzy již není považováno za bezpečné. Dalším srovnávaným algoritmem byla šifra AES. Tato šifra je všeobecně považována za bezpečnou, zvlášť v případě kombinace s hashovací funkcí SHA2-512 a klíčem o délce 256 bitů. Jako bezpečný je stále považován také algoritmus Blowfish, respektive jeho nástupce algoritmus Twofish. Jako nejsilnější testovanou kombinaci pak lze považovat algoritmus Serpent ve spojení s hashovací funkcí SHA2-512 a šifrovacím klíčem o délce 256 bitů.

V současné době je považován za bezpečný standard délka klíče alespoň 112 bitů. Rovněž platí, že s rostoucí délkou klíče se zvyšuje odolnost vůči brute-force útokům. S neustále se zvyšujícími výkonnostními limity počítačů tak je třeba používat klíče o větších délkách. Cenou za vyšší bezpečnost tak ale je zvýšená hardwarová náročnost, zvláště během fáze dešifrování. Delší klíče tak mají přednost v případech, kdy požadavky na míru zabezpečení přesahují potřeby rychlé interactivity. Příkladem mohou být portály bankovních a finančních institucí, či vládní instituce a systémy.

11 ZÁVĚR

Podle očekávání se ukázala problematika cloudových úložišť jako velmi rozsáhlé a komplikované téma. Navzdory tomu se podařilo při seznamování s cloudovými technologiemi nalézt velké množství kvalitních dokumentů, textů a prací, které mi pomohly utvořit si ucelený obraz o cloudových řešeních. Na základě něj byl sestaven stručný popis jednotlivých typů cloudových distribucí a architektur. V souvislosti s typy cloudů byl také adresován zásadní rozdíl mezi jednotlivými typy vlastnictví cloudu.

Rovněž byl popsán bezpečnostní model užívaný současnými firmami používajícími cloudové technologie. V návaznosti na jednotlivé vrstvy bezpečnostního modelu byly uvedeny potencionální hrozby a obvyklé užívané techniky prevence. Jelikož bylo cílem diplomové práce vytvořit vlastní návrh cloudového zabezpečení, tak součástí práce je i rozsáhlá analýza rizik a popis nejčastějších útoků vedených vůči cloudovým řešením. Dalším předmětem zkoumání se staly používané metody přenosu mezi uživatelem a cloudem. Popsány tak byly dva nejčastěji používané standardy připojení ke cloudovým řešením.

Hlavním cílem práce pak byla realizace cloudového úložiště na základě vlastního návrhu. Samotný návrh byl vytvořen na základě analýzy rizik, předchozí zkušenosti a dostupných možností. Návrh tak představuje kombinaci veřejně dostupných technologií a infrastruktury nabízené profesionálním cloudovým poskytovatelem. Důležitým aspektem návrhu byl také výběr technického řešení datové úspornosti navrhovaného systému. Požadavkem byla dostupnost a jednoduchost implementace. Jako řešení, nejlépe vyhovujícím těmto kritériím, bylo vybráno použití veřejně dostupného single-instance úložiště. Tento typ úložiště používá speciální komprimaci dat, která nahrazuje duplikátní data referencemi, čímž dochází k požadované úspoře využití kapacity disku.

Tato úspora výpočetních zdrojů byla navíc umocněna způsobem, jakým poskytuje cloudový provozovatel virtuální stroje svým zákazníkům. V rámci této práce tak jsou veškeré výpočetní zdroje alokovány „tence“, což v praxi znamená, že množství alokovaných zdrojů odpovídá aktuálně používanému množství výpočetních zdrojů.

Dalším cílem této práce bylo zajistit bezpečnost cloudového úložiště a zajištění bezpečného a snadného přístupu uživatelů k serverům úložiště. Samotné zabezpečení tak bylo realizováno, jak na přístupovém bodě cloudového řešení pomocí firewallu, tak na samotných serverech. Současně se podařilo implementovat pokročilou autentizaci uživatelů pomocí serveru RADIUS. Přístup uživatelů pak byl realizován pomocí technologie VPN a překladu adres NAT.

Zásadním rozdílem oproti návrhu pak bylo rozhodnutí užití IPSec tunelů stavěných programem OpenSwan namísto užití GRE tunelů poskytovaných operačním systémem. Toto rozhodnutí bylo učiněno na základě požadavku otestovat a ohodnotit efektivitu a šifrovací sílu dostupných šifrovacích algoritmů. Díky této změně bylo možné simulovat bezpečný šifrovaný přenos pomocí algoritmů 3DES, AES, CAST, Serpent. Z důvodu částečné nekompatibility operačního systému a programu OpenSwan nebylo možné experimentovat s algoritmem Blowfish. Namísto tohoto algoritmu jsem se rozhodl otestovat dostupný šifrovací algoritmus Twofish, jenž je nástupcem algoritmu Blowfish.

Na základě získaných výsledků a dostupných informací jsem provedl srovnání algoritmů z hlediska dosahovaných přenosových rychlostí a šifrovací síly. V závěru práce

jsem se věnoval experimentování s výše zmíněnými algoritmy a závislostí na délce použitých klíčů. Výsledky těchto testů potvrdily předpokládaný vztah nepřímé úměry mezi délkou šifrovacího klíče a rychlostí přenosu.

Jako důležité vylepšení takto řešeného cloudového úložiště bych považoval oddělení autentizačního serveru od datového úložiště. Dalším potencionálním vylepšením by mohl být překlad používaných portů na porty nestandardní. Z hlediska testování algoritmu by bylo možné rovněž sledovat podrobnější statistiky o prováděných operacích při navazování šifrovaného spojení, ustavování tunelu a při samotném přenosu. Na základě těchto výsledků provést další porovnání algoritmů.

Práce na této realizaci mi umožnila osvojit si velké množství používaných technologií a ucelit si představu o cloudových řešeních a současných trendech síťového zabezpečení. Tyto nabité zkušenosti a vědomosti budou zajisté cenným přínosem pro výkon mého povolání.

LITERATURA

[1] CROSMAN, Penny. Cloud Computing Begins to Gain Traction on Wall Street. *Wall Street & Technology, January*, 2009, 6. <http://www.wallstreetandtech.com/infrastructure/cloud-computing-begins-to-gain-traction-on-wall-street/d/d-id/1261032>

[2] BERGAMO, A. Calculating Cloud Computing Costs – CapEx vs. OpEx. *Hosting* [online], <http://www.hosting.com/computing-cloud-costs-capex-vs-opex>

[3] FOLEY, John. Private clouds take shape. *InformationWeek*, 2008. <http://www.informationweek.com/private-clouds-take-shape/d/d-id/1070793>

[4] KASSNER, M., Dedicated network connections improve hybrid-cloud performance and security. *TechRepublic*, june 2014, <http://www.techrepublic.com/article/dedicated-network-connections-improve-hybrid-cloud-performance-and-security/>

[5] MELL, Peter; GRANCE, Tim. The NIST definition of cloud computing. 2011.

[6] RUEST, Danielle; RUEST, Nelson. *Virtualizace: podrobný průvodce*. Computer Press, 2010.

[7] VELTE, Toby; VELTE, Anthony; ELSENPETER, Robert. *Cloud computing, a practical approach*. McGraw-Hill, Inc., 2009.

[8] CHOU, Timothy. Introduction to cloud computing business & technology. *Lecture Notes at Stanford University and at Tsinghua University, Active Book Press*, 2010.

[9] JAMSA, Kris. *Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More*. Jones & Bartlett Publishers, 2011.

[10] WINKLER, Vic JR. *Securing the Cloud: Cloud computer Security techniques and tactics*. Elsevier, 2011.

[11] THEUNS, M., Layering information security controls. CapGemini [online], 2015, <https://www.capgemini.com/blog/capping-it-off/2015/08/layering-information-security-controls>

[12] SARNA, David EY. *Implementing and developing cloud computing applications*. CRC Press, 2010.

[13] SOLOMON, Michael G.; CHAPPLE, Mike. *Information security illuminated*. Jones & Bartlett Learning, 2005.

[14] NORTHCUTT, Stephen, et al. *Inside network perimeter security: the definitive guide to firewalls, VPNs, routers, and intrusion detection systems*. Pearson Education, 2002.

[15] MILLER, Michael. *Cloud computing: Web-based applications that change the way you work and collaborate online*. Que publishing, 2008.

[16] PERRIN, Chad. The CIA triad. 2008 Dostupno z: <http://www.techrepublic.com/blog/security/the-cia-triad/488>

[17] HULBOJ, Milosz Marian; JURGA, Ryszard Erazm. Packet Sampling and Network Monitoring. 2007.

[18] FRUHWIRTH, Clemens. LUKS On-Disk Format Specification Version 1.2. 2011.

[19] CROSBY, S., et al. Open virtualization format specification. *Standards and Technology, no. DSP0243 in DMTF Specifications, Distributed Management Task Force*, 2009.

[20] PATEL, Pankesh; RANABAHU, Ajith H.; SHETH, Amit P. Service level agreement in cloud computing. 2009.

[21] CATTEDDU, Daniele. Cloud Computing: benefits, risks and recommendations for information security. In: *Web Application Security*. Springer Berlin Heidelberg, 2010. p. 17-17.

[22] LAU, Felix, et al. Distributed denial of service attacks. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*. IEEE, 2000. p. 2275-2280.

[23] SQALLI, Mohammed H.; AL-HAIDARI, Fahd; SALAH, Khaled. Edos-shield-a two-steps mitigation technique against edos attacks in cloud computing. In: *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011. p. 49-56.

[24] JENSEN, Meiko, et al. On technical security issues in cloud computing. In: *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*. IEEE, 2009. p. 109-116.

[25] LUO, Qiasi; FEI, Yunsi. Algorithmic collision analysis for evaluating cryptographic systems and side-channel attacks. In: *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*. IEEE, 2011. p. 75-80.

[26] MATYÁŠ, Václav, et al. Principy a technické aspekty autentizace. 2007.

[27] TYSON, Jeff. How Virtual private networks work. *Retrieved on July*, 2001, 31: 2008.

[28] FRIEDL, Steve. An illustrated guide to IPSec. *Unixwiz. net*, 2005.

[29] BOUTIN, Chad. NIST selects winner of Secure Hash Algorithm(SHA-3) Competition. *Press release., October*, 2012, 2.

[30] STEPHEN, Thomas. SSL and TLS Essentials. Securing the Web. 2000.

[31] ADAMS, Keith; AGESEN, Ole. A comparison of software and hardware techniques for x86 virtualization. *ACM Sigplan Notices*, 2006, 41.11: 2-13.

[32] Official Amazon Web pages, aws.amazon.com [online], <https://aws.amazon.com/ec2/>

[33] Official Navisite Web pages, navisite.com, [online], <http://www.navisite.com/services/cloud-infrastructure-services/self-service-cloud>

[34] Official NTT Web pages, ntt.com [online], <http://www.ntt.com/en/services/cloud/enterprise-cloud.html>

[35] PAL, Lipika. *VMware vCloud Director Essentials*. Packt Publishing Ltd, 2014.

[36] GULATI, Ajay, et al. Vmware distributed resource management: Design, implementation, and lessons learned. *VMware Technical Journal*, 2012, 1.1: 45-64.

[37] ALIBI, Mohamed; ROY, Bhaskarjyoti. *Mastering CentOS 7 Linux Server*. Packt Publishing Ltd, 2016

[38] Official CentOS Web pages, wiki.centos.org [online] <https://wiki.centos.org/About/Product>

[39] VAUGHAN, J. S., The most popular Linux for Web server is…, Computerworld.com [online] <http://www.computerworld.com/article/2468596/network-software/the-most-popular-linux-for-web-servers-is----.html>

[40] SINGH, K., CentOS Project joins forces with Red Hat [online], 2014 <https://lists.centos.org/pipermail/centos-announce/2014-January/020100.html>

[41] SRISURESH, Pyda; HOLDREGE, Matt. IP network address translator (NAT) terminology and considerations. 1999.

[42] REKHTER, Yakov, et al. Address allocation for private internets. 1994.

[43] BELLOVIN, Steven M. Distributed firewalls. *Journal of Login*, 1999, 24.5: 37-39.

[44] PURDY, Gregor N. *Linux iptables pocket reference*. O'Reilly Media, Inc., 2004.

[45] FEILNER, Markus. *OpenVPN: Building and integrating virtual private networks*. Packt Publishing Ltd, 2006.

[46] Official OpenVPN Web pages, openvpn.net [online], <https://openvpn.net/index.php/component/content/article/64-access-server-paid/general/515-release-notes-v183.html>

[47] LOCKHART, Andrew. *Network security hacks*. O'Reilly Media, Inc., 2006.

[48] Official OpenVPN Web pages, openvpn.net [online], <https://community.openvpn.net/openvpn/wiki/RelatedProjects>

[49] Official FreeRADIUS Web pages, freeradius.org [online], <http://freeradius.org/press/survey.html>

[50] MYSQL, A. B. MySQL. 2001.

[51] SIMPSON, William Allen. PPP challenge handshake authentication protocol (CHAP). 1996.

[52] LLOYD, Brian; SIMPSON, William. PPP authentication protocols. 1992.

[53] Official OpenSwan Web pages, openswan.org [online], <https://www.openswan.org/>

[54] BOWLING, Jeramiah. Opendedup: open-source deduplication put to the test. *Linux Journal*, 2013, 2013.228: 2.

[55] VEPRINSKY, Alexandr; MICHAEL, Ofer E.; SCHARLAND, Michael J. *Data de-duplication using thin provisioning*. U.S. Patent No 7,822,939, 2010.

[56] ROSEN, Rami. Creating vpns with ipsec and ssl/tls. *Linux Journal*, 2008, 2008.165: 10.

[57] BHANOT, Rajdeep; HANS, Rahul. A Review and Comparative Analysis of Various Encryption Algorithms. *International Journal of Security and Its Applications*, 2015, 9.4: 289-306.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

MD5	Message Digest Algorithm
TCP	Transmission Control Protocol
VPN	Virtual Private Network
SSL	Secure Sockets Layer
TLS	Transport Layer Security
IPSec	Internet Protocol Security
SHA	Secure Hash Algorithm
3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
IKE	Internet Key Exchange
MAC	Message authentication code
INA	Internet Network Appliance
ESP	Encapsulating Security Payload
NAT	Network Address Translation
API	Application Programming Interface
REST	Representational State Transfer
OVF	Open Virtualization Format
AH	Authentication Header
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
RAM	Random Access Memory
CPU	Central Processing Unit

PŘÍLOHA 1

Pomocný spouštěcí skript IPSec start_ipsec.sh

```
service ipsec start
ipsec auto --add mytunnel
wait 10
ipsec auto --up mytunnel
service ipsec status
```

Pomocný ukončovací skript IPSec stop_ipsec.sh

```
ipsec auto --delete mytunnel
service ipsec stop
service ipsec status
```

Testovací skript – spouštění testu count.sh

```
time ./copy.sh
time ./copy.sh
time ./copy.sh
time ./copy.sh
time ./copy.sh
```

Kopírovací skript –copy.sh

```
START_TIME=$SECONDS
echo $START_TIME
scp output1.dat 10.0.0.146:/tmp/output1.dat
scp output2.dat 10.0.0.146:/tmp/output2.dat
scp output3.dat 10.0.0.146:/tmp/output3.dat
scp output4.dat 10.0.0.146:/tmp/output4.dat
scp output5.dat 10.0.0.146:/tmp/output5.dat
scp output6.dat 10.0.0.146:/tmp/output6.dat
scp output7.dat 10.0.0.146:/tmp/output7.dat
scp output8.dat 10.0.0.146:/tmp/output8.dat
scp output9.dat 10.0.0.146:/tmp/output9.dat
scp output10.dat 10.0.0.146:/tmp/output10.dat
ELAPSED_TIME=$(($SECONDS - $START_TIME))
echo $ELAPSED_TIME
ssh root@10.0.0.146 'rm -f /tmp/output*'
```

PŘÍLOHA 2

Fáze 2 (128 bit)			Naměřené výsledky						
Šifrovací algoritmus	Hashovací funkce	Diffie Hellman Group	Průměrná doba přenosu dat [s]	Průměrná doba přenosu [MB/s]	Pokus 1	Pokus 2	Pokus 3	Pokus 4	Pokus 5
3DES	MD5	modp1024	9.13998	10.94094298	9.1837	9.1678	9.114	9.1202	9.1142
3DES	SHA1	modp1024	9.69168	10.31812854	9.7002	9.6969	9.6554	9.7289	9.677
3DES	SHA2_512	modp1024	9.59224	10.42509362	9.6439	9.5896	9.5726	9.5868	9.5683
CAST	MD5	modp1024	3.63982	27.47388607	3.6259	3.5908	3.6066	3.6318	3.744
CAST	SHA1	modp1024	4.1665	24.00096004	4.1987	4.1833	4.1743	4.1356	4.1406
CAST	SHA2_512	modp1024	3.97626	25.14926086	3.9295	3.9895	4.0033	3.9601	3.9989
BLOWFISH	MD5	modp1024	0						
BLOWFISH	SHA1	modp1024	0						
BLOWFISH	SHA2_512	modp1024	0						
AES	MD5	modp1024	2.34792	42.59088896	2.317	2.3597	2.4211	2.3227	2.3191
AES	SHA1	modp1024	2.86582	34.89402684	2.8562	2.883	2.9224	2.8282	2.8393
AES	SHA2_512	modp1024	2.8362	35.2584444	2.8223	2.8442	2.8541	2.8258	2.8346
SERPENT	MD5	modp1024	4.22022	23.69544716	4.2167	4.2025	4.1954	4.2562	4.2303
SERPENT	SHA1	modp1024	4.7272	21.1541716	4.754	4.6613	4.7461	4.736	4.7386
SERPENT	SHA2_512	modp1024	4.51932	22.12722268	4.4796	4.524	4.5127	4.5146	4.5657
TWOFISH	MD5	modp1024	3.08342	32.43152084	3.1453	3.0959	3.0951	3.0446	3.0362
TWOFISH	SHA1	modp1024	3.55188	28.15410431	3.5423	3.539	3.5809	3.5643	3.5329
TWOFISH	SHA2_512	modp1024	3.368	29.6912114	3.3687	3.3184	3.3699	3.3591	3.4239

Fáze 2 (256 bit)			Naměřené výsledky						
Šifrovací algoritmus	Hashovací funkce	Diffie Hellman Group	Průměrná doba přenosu dat [s]	Průměrná doba přenosu [MB/s]	Pokus 1	Pokus 2	Pokus 3	Pokus 4	Pokus 5
3DES	MD5	modp1024	0		Podporován je pouze klíč o délce 192 bitů				
3DES	SHA1	modp1024	0						
3DES	SHA2_512	modp1024	0						
CAST	MD5	modp1024	0		Podporován je pouze klíč o délce 192 bitů				
CAST	SHA1	modp1024	0						
CAST	SHA2_512	modp1024	0						
BLOWFISH	MD5	modp1024	0						
BLOWFISH	SHA1	modp1024	0						
BLOWFISH	SHA2_512	modp1024	0						
AES	MD5	modp1024	2.52684	39.5751215	2.487	2.4854	2.5057	2.5799	2.5762
AES	SHA1	modp1024	2.96374	33.74115138	3.0328	2.943	2.947	2.9425	2.9534
AES	SHA2_512	modp1024	2.99844	33.35067568	2.9813	2.9918	3.0272	2.9811	3.0108
SERPENT	MD5	modp1024	4.201	23.80385622	4.215	4.2007	4.1699	4.1991	4.2203
SERPENT	SHA1	modp1024	4.7526	21.04111434	4.7299	4.763	4.7647	4.7929	4.7125
SERPENT	SHA2_512	modp1024	4.5379	22.03662487	4.5985	4.5031	4.5269	4.5246	4.5364
TWOFISH	MD5	modp1024	2.9996	33.33777837	3.0019	2.9962	3.0074	2.9911	3.0014
TWOFISH	SHA1	modp1024	3.6089	27.70927429	3.6899	3.6528	3.516	3.6803	3.5055
TWOFISH	SHA2_512	modp1024	3.41534	29.27966176	3.4377	3.4073	3.4305	3.3811	3.4201